

NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



THESIS

TRANSIENT FIELD VISUALIZATION FOR ULTRA- WIDEBAND ANTENNA DESIGN

by

David V. Adamiak

December 1999

Thesis Advisor:
Second Reader:

Michael A. Morgan
R. Clark Robertson

Approved for public release; distribution is unlimited.

20000309 017

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.</p>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1999		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE TRANSIENT FIELD VISUALIZATION FOR ULTRA-WIDEBAND ANTENNA DESIGN			5. FUNDING NUMBERS	
6. AUTHOR(S) David V. Adamiak				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
<p>13. ABSTRACT (maximum 200 words)</p> <p>Impulse antennas are specifically designed to transmit and/or receive very short bursts of electromagnetic energy. By their very nature, these short time-duration bursts (impulses) require ultra-wideband transmitting and receiving antennas. This thesis investigates a number of UWB antenna designs to determine their feasibility in receiving an impulse having a 1000:1 bandwidth (10 MHz to 10 GHz) with virtually no distortion.</p> <p>As a tool in aiding the design of such an antenna, this thesis presents original software that was developed to visualize an impulse propagating in the near-field region of the antenna being considered. Such software will significantly reduce the workload and time required for antenna design and provide unique capabilities for heuristic understanding of the physics involved.</p>				
14. SUBJECT TERMS Impulse Antenna, Wideband Antenna, Ultra-wideband, Near-field, Software			15. NUMBER OF PAGES 145	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18 298-102

Approved for public release; distribution is unlimited.

TRANSIENT FIELD VISUALIZATION FOR ULTRA-WIDEBAND ANTENNA DESIGN

David V. Adamiak
Major, United States Marine Corps
B.S., United States Naval Academy, 1985

Submitted in partial fulfillment
of the requirements for the degree of

**MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING**

from the

NAVAL POSTGRADUATE SCHOOL

December 1999

Author:

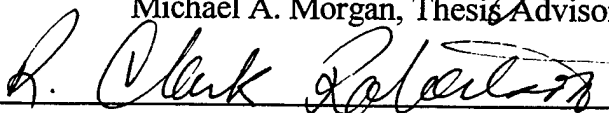


David V. Adamiak

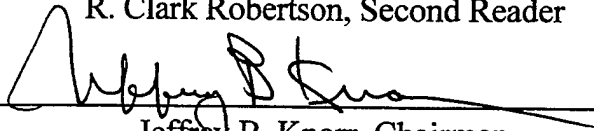
Approved by:



Michael A. Morgan, Thesis Advisor



R. Clark Robertson, Second Reader



Jeffrey B. Knorr, Chairman

Department of Electrical and Computer Engineering

ABSTRACT

Impulse antennas are specifically designed to transmit and/or receive very short bursts of electromagnetic energy. By their very nature, these short time-duration bursts (impulses) require ultra-wideband transmitting and receiving antennas. This thesis investigates a number of UWB antenna designs to determine their feasibility in receiving an impulse having a 1000:1 bandwidth (10 MHz to 10 GHz) with virtually no distortion.

As a tool in aiding the design of such an antenna, this thesis presents original software that was developed to visualize an impulse propagating in the near-field region of the antenna being considered. Such software will significantly reduce the workload and time required for antenna design and provide unique capabilities for heuristic understanding of the physics involved.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. ULTRA-WIDEBAND ANTENNA THEORY	3
A. BACKGROUND	3
B. IMPULSE RECEIVING ANTENNAS	6
C. TEM HORN ANTENNA	9
D. LORENTZ RECIPROCITY THEOREM	13
III. ANALYSIS OF WIRE ANTENNAS AND SCATTERERS (<i>AWAS</i>)	15
A. BACKGROUND	15
B. DATA INPUT	17
C. INPUT AND OUTPUT FILES	21
D. PROGRAMMING NOTES	26
IV. VISUALIZATION OF A PROPAGATING IMPULSE	29
A. SOFTWARE OVERVIEW	29
B. INITIALIZING THE SOFTWARE	31
C. DATA INPUT & CALCULATIONS	33
D. OUTPUT FILES	37
E. GRAPHIC DISPLAYS	39
F. MOVIE SUBROUTINE	42
V. VALIDATION OF THE SOFTWARE AND ANALYSIS OF THE PROTOTYPE	
TEM HORN	49
A. SIMPLE WIRE ANTENNA STRUCTURES	49
1. A Simple Dipole	50
2. The Vee Dipole	65
3. The Vee Dipole with a Parallel Extension	69
4. The Vee Dipole with Parallel Extension Terminated at a $1k\Omega$ Load	72
B. TEM HORN PROTOTYPE DESIGN	76

VI. CONCLUSIONS.....	87
LIST OF REFERENCES.....	89
APPENDIX A - NEAR-FIELD VISUALIZATION SOFTWARE (NFLDVIS.M).....	91
APPENDIX B - MATLAB CODE.....	123
INITIAL DISTRIBUTION LIST	127

LIST OF FIGURES

Figure 1. Example of a TEM Horn	9
Figure 2. Example of Wire Antenna	16
Figure 3. Main <i>AWAS</i> Window.....	17
Figure 4. <i>AWAS</i> Input Data & Show Windows at Start-Up	18
Figure 5. Example of <i>AWAS</i> Data Input & Antenna Display.....	19
Figure 6. Example of <i>AWAS</i> Antenna Geometry (.GEO) Input File	22
Figure 7. Example of <i>AWAS</i> Near-Field (.NFL) Output File.....	24
Figure 8. Example of <i>AWAS</i> Near-Field (.NFL) Output File.....	26
Figure 9. Initial Window on Start-Up of NFLDVIS.M.....	32
Figure 10. Selection of .NFL File for Analysis	34
Figure 11. Gaussian Impulse Input Voltage	36
Figure 12. $ V_{in}(f) $, The Frequency Spectrum of $V_{in}(t)$	36
Figure 13. Example of "Filtered" Frequency-Domain E-Field	37
Figure 14. Example of E-Field Frequency-Domain Plot	39
Figure 15. Example of 3-D Plot As View Angle Is Altered.....	41
Figure 16. Example of Gaussian Input Voltage Plot.....	42
Figure 17. Example of Initial Sample Frame from Movie Subroutine	44
Figure 18. Sample Frame from Movie after Modification of Z-Axis and View Angle	45
Figure 19. Sample Frame from Movie after Modification of the C-Axis.....	46
Figure 20. A Half-Wave Dipole and Its Current Distribution.....	50
Figure 21. Radiation Pattern for an Ideal Dipole.....	51
Figure 22. Geometry Used in Dipole E-Field Calculations. From [8].....	52
Figure 23. The Theoretical Near Electric Field, $ E_z(y,f) $, of a Simple Dipole	53
Figure 24. The Theoretical Interference Pattern in the Near Field of the Simple Dipole.....	54
Figure 25. Plot of E-Field Magnitude vs. Frequency and Position Along the Y-Axis for the Simple Dipole	55

Figure 26. Plot of E-Field Magnitude as a Function of Distance vs. Frequency	56
Figure 27. Comparison Between AWAS vs. Theoretical $ E_z $ for a $\lambda/4$ Dipole	57
Figure 28. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $\lambda/4$ Dipole	57
Figure 29. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $\lambda/4$ Dipole	58
Figure 30. Comparison Between AWAS vs. Theoretical $ E_z $ for a $\lambda/2$ Dipole	59
Figure 31. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $\lambda/2$ Dipole	59
Figure 32. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $\lambda/2$ Dipole	60
Figure 33. Comparison Between AWAS vs. Theoretical $ E_z $ for a $3\lambda/2$ Dipole	60
Figure 34. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $3\lambda/2$ Dipole	61
Figure 35. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $3\lambda/2$ Dipole	61
Figure 36. A Snapshot of the Time-Domain E-Field of a Simple Dipole Transmitter at $t=13.5\text{nsec}$	62
Figure 37. Incident E-Field Generated by a Double-Gaussian Impulse at $t=15\text{nsec}$	63
Figure 38. The Time-Domain E-Field of a Simple Dipole Receiver at $t=20.5\text{nsec}$	64
Figure 39. Example of a Vee Dipole	65
Figure 40. Plot of E-Field Magnitude vs. Frequency and Distance along the Centerline Axis for the Vee Dipole	66
Figure 41. The E-Field along the Vee Dipole's Centerline Axis	67
Figure 42. A Snapshot of the Time-Domain E-Field of the Vee Dipole	68
Figure 43. Example of a Vee Dipole with Parallel Extension	69
Figure 44. The Frequency-Domain E-Field for the Vee Dipole with Parallel Extension	70
Figure 45. A Different Perspective of the Plot Shown in Figure 44	71
Figure 46. The Time-Domain E-Field of the Vee Dipole with Parallel Extension	72
Figure 47. Vee Dipole with Extension Terminated at a $1\text{k}\Omega$ Load	73
Figure 48. Frequency-Domain E-Field of the Extended Vee Dipole Terminated at a $1\text{k}\Omega$ Load	74
Figure 49. Time-Domain E-Field for the Extended Vee Dipole Terminated at a $1\text{k}\Omega$ Load	75

Figure 50. 3-D and Overhead Views of the Prototype TEM Horn	76
Figure 51. Frequency-Domain Plot of the E-Field for the Prototype TEM Horn	77
Figure 52. Frequency Response of the TEM Horn at the Feed Point	78
Figure 53. Frequency-Domain Plot of the TEM Horn's E-Field with Area of Interest Highlighted	79
Figure 54. Time-Domain E-Field for the Prototype TEM Horn.....	80
Figure 55. Time-Domain E-Field for the Prototype TEM Horn.....	81
Figure 56. Frequency-Domain E-Field for the TEM Horn at Low Frequencies	82
Figure 57. Prototype TEM Horn E-Field Showing Possible Resonant Frequency	82
Figure 58. Close-up of Possible Resonant Frequency	83
Figure 59. Incident Double-Gaussian Impulse E-Field	84
Figure 60. Time-Domain Reflected E-Field within the TEM Horn Receiver	84
Figure 61. Resultant Transmitted and Reflected Fields at the TEM Horn Aperture	85

LIST OF TABLES

1. Upper Limits for All <i>AWAS</i> Versions.....	27
---	----

ACKNOWLEDGMENT

The author would like to thank Prof. Michael Morgan for his guidance and inspiration through every step of this research, Mr. Colin Cooper for his aid in understanding the nuances of programming, and most importantly, my wife, Cristina, and son, Andrew, whose patience, encouragement, and love made all the difference.

I. INTRODUCTION

Antennas have been a part of our lives since most of us can remember. Whether the simple car whip antenna, the old Ultra-High Frequency (UHF) "rabbit ears," or the television antenna that seemed to blow off the roof every winter, we have come to accept them as a normal part of life. Within the military, our daily use of antennas is a constant which causes us to take for granted the process of optimizing an antenna design to the specific task/weapon system. Truly, we are accustomed to seeing the effects of an antenna, and never the causes for what we see on the screen or hear through a speaker. It is only when the effects are not what we desire that we may actually think of the antenna as an integral part of the system.

As technology continues to move forward, though, there are increasing applications of very specialized antennas for weapon systems and communications, alike. This work centers around just such an application. Specifically, it stems from a requirement to design an antenna that is capable of receiving and accurately reproducing a short-duration impulse of electromagnetic energy, or an impulse receiving antenna. From signal analysis theory, it can be shown that a single pulse of electromagnetic energy that is of extremely short time duration (an impulse) will infer a frequency characteristic of very wide bandwidth [1]. Specifically, if one were able to create a pulse of infinitely short duration, the resulting frequency bandwidth of that impulsive signal would be infinitely wide. In reality, this is impossible; however, that inverse relationship between signals in the time- and frequency-domain is still valid. For that reason, alone, if we look to design and construct an antenna which is able to transmit or receive such a short duration, impulsive signal, then we must take into consideration the requirement for the antenna to operate flawlessly across that entire ultra-wide bandwidth.

This research will investigate the theory behind ultra-wideband (UWB) antennas and develop a prototype design of an UWB antenna that is capable of receiving and replicating an impulse at near-perfect fidelity over the frequency range of 10 MHz to 10

GHz (1000:1 bandwidth ratio). While there exists a number of antennas which are advertised to be ultra-wideband in nature, the reality is that most do not have the linear phase characteristic necessary for replication of time-domain signals. Specific examples (and their maximum bandwidths) are spiral antennas (50:1), quadrature antennas (36:1), and log-periodic antennas (40:1).

In addition, while there are certainly numerous “tools” available to the antenna designer, the unique requirement to preserve the fidelity of the signal in the time domain poses the question: How does one best acknowledge an antenna’s performance in the time domain without actually analyzing it there? To that end, the time-domain representation of an antenna and its near electromagnetic fields (near fields) is, to say the least, a dynamic environment. But the requirement to measure performance in the time domain is a necessity in determining the ability of such an antenna to meet its design specifications.

As such, a specific goal within the design process, and the primary focus of this work, is the creation, testing, and application of software that would aid in the design process through a visual display of the dynamic near fields in the time domain. Through this, an individual of even minimal background in antenna design could then be afforded the opportunity to “see and learn,” to observe the changes in the near fields, and heuristically understand the physics of what is occurring.

To begin, we will first discuss the theory behind UWB antennas, explore the possible designs which would fulfill our requirement, and choose a prototype design for analysis. Following that, this work will explain in detail the combination of the *Analysis of Wire Antennas and Scatterers (AWAS)* and *Near-Field Visualization (NFLDVIS.M)* software packages for use in the visualization and animation of the antenna’s near fields. From there, this work will trace the validation/verification of the software as an accurate method for displaying the near fields, and then conclude with the actual analysis of a prototype UWB antenna design.

II. ULTRA-WIDEBAND ANTENNA THEORY

As stated earlier, our search for an antenna that is able to accurately reproduce the time-domain waveform of a received impulse signal requires us to look for an antenna that is able to operate over a very wide range of frequencies. The shorter the time duration of the impulse waveform, the wider the frequency range over which our antenna must operate. Our discussion, now, turns to the definition of wideband and the antennas that truly meet this definition.

A. BACKGROUND

A broadband, or wideband, antenna is generally defined as one whose impedance and pattern “do not change significantly over about an octave ($f_U/f_L = 2$)” [2], where f_L and f_U are the lower and upper limits on the bandwidth, or frequency range, respectively. Using this definition, then, it is easy to see that our desired bandwidth ratio ($f_U/f_L = 10\text{GHz}/10\text{MHz} = 1000$) implies a need for an ultra-wideband antenna. In general, a typical wideband antenna must be a structure that exhibits a smoothness throughout its dimensions and does not tend to have abrupt physical changes, even at the boundaries. This structure will then produce constant, smooth field amplitudes and input impedances that either remain unchanged or change linearly (smoothly) with frequency [2].

Long-wire antennas are well known as having excellent broadband properties, especially in the High Frequency (2 MHz to 30 MHz) range. As the name implies, these antennas are simply long wires supported by towers at either end. They may be operated as either resonant antennas, where the current on the wire will be in the form of a standing wave, or non-resonant (traveling wave) antennas. The physical difference between the two is the introduction of a matched load at the end of the non-resonant long-wire antenna which creates a traveling current wave in much the same manner as a transmission line terminated at a matched load. Also, the traveling-wave antennas are

more directive, lacking the presence of a major lobe in the reverse direction of antenna orientation. As an example, a long-wire antenna oriented along the positive z -axis will radiate main lobes in both the positive and negative z -direction if the antenna is a resonant one, and only in the positive direction if it is non-resonant.

The traveling-wave antennas have an advantage over their resonant cousins in that their input impedance is predominantly resistive (real), just as the impedance of a pure traveling wave on a low-loss transmission line is equal to the characteristic, real impedance of that line. Because of this, and the fact that the traveling-wave antenna's input impedance is also relatively independent of frequency, these non-resonant antennas will result in both an improved radiation pattern and a wider bandwidth as compared to the equivalent resonant structure. [2]

One drawback to both types of long-wire antennas is their size. An antenna designed for a 10 MHz operating frequency would be at least 15 meters long ($\lambda_o/2$), and can easily exceed six wavelengths in length. As a result, these antennas can in no way be considered even remotely portable, or trainable in any but one direction. A second drawback is that except for those frequencies where the antenna's main lobe is nearly parallel to the antenna structure, there will always be a portion of the radiation reflected off the ground. This reflected field is then out of phase with the direct radiation. For a receiver, the reflected field energy will have a phase delay in comparison with the direct field. Although this does not create much of a problem for normal continuous wave (CW) applications, we will see that a phase delay will have significant impact upon an impulse receiving antenna. Lastly, although the input impedance is frequency independent, it is very much dependent upon antenna length, and so the longer the antenna, the greater input impedance (resistance) required. This serves to severely restrict the upper frequency limit, f_U , to less than 1 GHz.

Other broadband antennas, such as helical antennas, biconical antennas, sleeve antennas and spiral antennas can generate larger bandwidth ratios than the long-wire antennas; however, they tend to suffer problems of scale when the bandwidth ratio begins

to reach 20:1. For example, the spiral antenna, which is typically referred to as a frequency independent antenna due to the fact its radiation pattern and impedance remain constant over a bandwidth ratio of 10:1 or higher, is “designed to minimize finite lengths and maximize angular dependence” [2]. However, its outer radius, R , is still equal to $\lambda_L/4$, or 7.5 meters for $f_L=10$ MHz. While providing a better bandwidth ratio than the long-wire antenna, at over 40 feet in diameter, it is even less portable than the long-wire antenna, and more difficult to construct. In addition, the phase response of the spiral antenna is highly nonlinear, resulting in dispersion of the time-domain pulse [4]. This, alone, makes the spiral antenna unsuitable as an impulse receiving antenna.

Another antenna with general wideband applications is the log-periodic antenna, which is seen quite often for television reception. These antennas tend to maintain simple geometrical shapes or involve merely straight wire segments, and thus are more easy to construct than the spiral antenna. The underlying concept in their construction is that the log-periodic antenna is scaled into itself periodically as the frequency (or wavelength) changes. This stems from the fact that an antenna which operates well at a wavelength λ_1 will operate just as well at a wavelength λ_2 if the antenna's dimensions are changed by a factor of λ_2/λ_1 . It can be shown that every dimension from one element to the next in the log-periodic antenna is related by a factor τ , which is called the log period. This can be determined simply from the fact that for a log-periodic antenna, $\log(f_2/f_1)=\log(\tau)$ [2], where $f_1=f_U$, and f_2 is the next lower design frequency where optimal performance is required.

This antenna, while providing greater bandwidth flexibility, still can be hindered by size as f_L approaches 10 MHz. In addition, the highest frequencies of a broadband pulse applied to a log-periodic antenna will occur first, and the lower frequencies will occur last when the antenna response is viewed in the time domain. This dispersion in the time-domain is driven by the poor phase response of the antenna. Like the highly dispersive spiral antenna, the nonlinear frequency response of the log-periodic antenna has little

impact on CW operations, but will be detrimental to the antenna's operation as an impulse receiver.

B. IMPULSE RECEIVING ANTENNAS

Traveling-wave wire antennas, and helical, spiral, and log-periodic antennas are but a few of the antennas which support broadband CW applications. However, when the signal changes from a CW one to an impulse, we find the dynamics of the antenna design process can change, too. Specifically, there tends to arise additional parameters that must be taken into consideration when an antenna is no longer being used for CW operations. As previously discussed, the requirement driving this research is to design an UWB antenna to be used for receiving time-domain signals with near-perfect fidelity. Thus, it is extremely critical that the shape of the impulse waveform be preserved throughout the entire length of the antenna. Specifically, then, this antenna will have an additional requirement to maintain a linear phase change across the entire frequency range. This is because if any group of frequencies is either favored or rejected over the others, the time-domain waveform as seen by the antenna will be distorted in comparison to the original incident waveform [3]. From [1], it can be shown that for any signal, a delay in the time domain is caused by a frequency shift in the frequency domain. In the case of a single Gaussian impulse, a frequency shift in one group of frequencies compared to the rest will cause a time delay in those frequency components, only. This will then cause dispersion in the output waveform. If this frequency shift (phase change) is linear across all frequencies, however, the entire signal will be delayed in time, and the precise shape of the original waveform (signal) will be left intact.

Additionally, since our antenna is intended to only act as a receiver, the *effective length* of the antenna will prove to be critical. Effective length is determined by

$$L_{\text{eff}}(\omega) = \frac{V_{\text{oc}}(\omega)}{E^i(\omega)}, \quad (2.1)$$

where $\omega = 2\pi f$ is the frequency in radians, $V_{oc}(\omega)$ is the open circuit voltage produced at the terminals of an antenna with an electric field $E^i(\omega)$ incident upon it [2],[4]. This is a key design parameter for an UWB antenna because in order to ensure the output voltage waveform is a near-perfect replication of the original incident pulse, the effective length must remain constant through the entire frequency range. The relationship between gain, G , and L_{eff} is given by [4]

$$G = \frac{\eta_o \pi}{R_r} \left(\frac{L_{eff} f}{c} \right)^2, \quad (2.2)$$

where η_o is the intrinsic impedance of free space, f is the frequency in Hz, c is the speed of light, and R_r is the radiation resistance of the antenna. Through this relationship, it is clear that an UWB receiving antenna requiring a constant effective length across its entire bandwidth will not have a constant gain when the radiation resistance is held constant, too. Rather, the gain will be proportional to the square of the frequency, which is different from standard CW antennas and even impulse transmitting antennas.

All of the previously discussed antennas can severely distort the received waveform, predominantly due to the nonlinear phase change mentioned above. Because of this and their overall inflexible size constraints, more specialized antennas must be found that will give us the bandwidth ratio and level of fidelity we desire. Reference [4] is the most comprehensive exploration of all the possible UWB impulse receiving antennas which may meet our bandwidth ratio requirement. The remaining portion of this section will summarize the pros and cons for each antenna type investigated in [4].

Resistively loaded linear antennas are “similar in concept to the standard traveling-wave antennas except, instead of a single resistive termination at the end of the device, the antenna is continuously loaded along its entire length” [4]. Although not highly directional, a 14cm long dipole was found to exhibit little dispersion and achieve a relatively constant effective length from 100 kHz to 1 GHz; however, this antenna’s ability to reach frequencies higher than 1 GHz is hampered by the fact its input impedance is over 200Ω. Due to this, a definite impedance mismatch between the

antenna and load will exist. To compensate for this and ensure balanced operation, a transformer will be required. Since no transformer is currently capable of operating at frequencies above 1 GHz, any effort to scale the antenna for use within our desired bandwidth would surely be moot. [4]

An orthogonally polarized quadruphase antenna has the advantages of both low dispersion and a low Voltage Standing Wave Ratio (VSWR) over its bandwidth of 500 MHz to 18 GHz. It is also easy and inexpensive to build. However, it lacks a frequency independent effective length, which will most certainly yield distortion of the time-domain waveform. In addition, the same structural size dimensions required of the long-wire antennas to reach f_L as discussed earlier make this antenna impractical, too. [4]

Modified log-periodic antennas differ from standard log-periodic antennas in that their spacing from element to element is changed to produce a more linear phase response. This, in turn, provides for less distortion of the time-domain waveform, and better overall performance. Unfortunately, though, the size of the elements required to receive the lowest frequency components of the impulse signal are still on the order of a half-wavelength, so even modified, the log-periodic antenna is not a wise choice for this application. [4]

Short axial length broadband horns and reflector antennas with wideband impulse feeds also show promise as impulse receiving antennas and have demonstrated solid wideband applications [4]. As with nearly every antenna mentioned previously, though, these, too, suffer from size constraints required to receive the low frequency components. The fact that they can receive an ultra-wideband pulse effectively is far overshadowed by their required size to get the job done. Thus, when we impose any type of size restriction or directional/mobility requirement upon our antenna, it is clear that these antennas will not make "the cut."

The only known antenna type which can be classified as truly ultra-wideband and is both relatively easy to manufacture and somewhat portable is the TEM horn. Named after the fact that it allows the fields within its structure to remain in the transverse

electromagnetic (TEM) mode, the TEM horn combines constant, frequency-independent effective length, good directionality, and a simple design to provide near-perfect fidelity in the reception of impulsive signals [3]. It has, without question, the best potential for meeting the design requirements previously set forth.

C. TEM HORN ANTENNA

The TEM horn, as illustrated in Figure 1, consists of two conductive triangular plates which gradually slope along an angle of no more than ten degrees, and then are attached to two resistively-loaded parallel plates. For the prototype that will be analyzed later, the exact dimensions are 72in x 6in x 4in (1.8288m x 0.1524m x 0.1026m). The triangular sections extend 36in (0.9144m) from the feed point along the y -axis, and the included angle between them is approximately 9.5° . The parallel plates are 36in x 4in (0.9144m x 0.1026m), and have resistive loads running parallel to the antenna boresight axis (the positive y -direction). These resistive loads increase roughly exponentially with distance from the flare-parallel-plate interface. It should be apparent that the TEM horn can be modeled, in a very approximate sense, as a parallel plate transmission line.

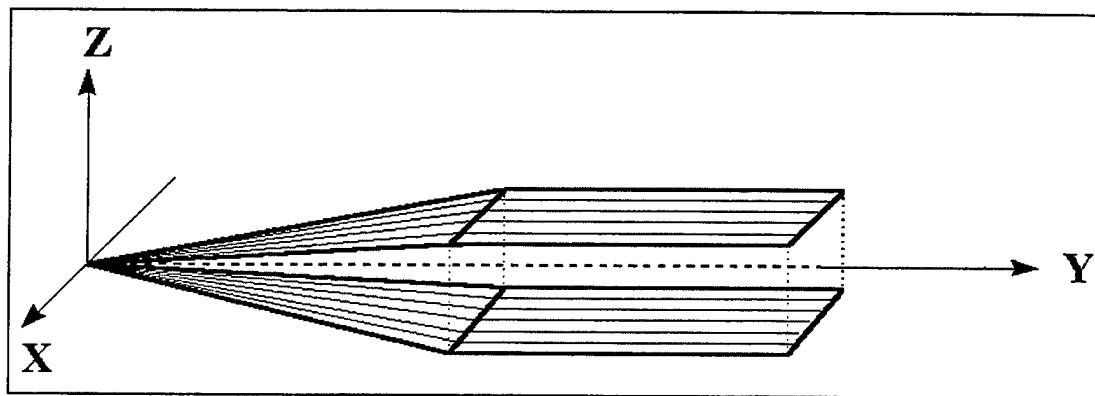


Figure 1. Example of a TEM Horn

To discover the benefits of the TEM horn in UWB antenna design, we first turn our attention to the concepts behind its development. As stated earlier, the time-domain antenna is constructed to accurately reproduce the time waveform of the incoming

electromagnetic fields. Unfortunately, most common receiving antennas severely distort these incident fields, allowing any number of modes to propagate from a single incident wave. These modes, in turn, cause undesirable distortion in the received waveform, restricting the operational bandwidth where accurate fidelity can be maintained. The TEM horn is designed to minimize these modes, and thus minimize distortion by allowing the electromagnetic fields within the structure to remain in the transverse electromagnetic (TEM) mode.

To provide for near-perfect fidelity in the reproduction of a received waveform, it is critical that the antenna maintain a constant (or linear) phase response. To accomplish this, the TEM horn is constructed so that the currents induced in the conductors by the incident fields move in the same direction as the fields, themselves. Because of this, the induced currents and incident fields stay in good time synchronization, thereby preserving the phase response.

Contrarily, a long dipole antenna impinged by those same fields has induced currents which would move perpendicular to the direction of propagation. This leads to a time lag between arrival of the incident fields and the induced currents at the dipole's feed point. Because of this time lag, the resultant waveform (taken as output from the feed) would exhibit dispersion, and thus distortion, in comparison to the original incident wave.

An additional phase shift can result from a time delay caused by the difference in path length between the conductors along the centerline axis of the TEM horn and those along the outside edge of the antenna. To minimize any potential time delay (phase change) caused by the longer outside path length, the TEM horn is always several times longer than it is wide. Additionally, the two triangular plates of the TEM horn depicted in Figure 1 are always very acute-angled and isosceles. As long as the path length difference is less than one-half wavelength for all operating frequencies, then the resultant phase shift will be negligible. To find the frequency above which this will present a

problem for our prototype TEM horn, we must first determine the path length difference from

$$\Delta l \approx l \left[\sqrt{1 + (W/2l)^2} - 1 \right], \quad (2.3)$$

where l is the length of the triangular plates, and W is the width of the TEM horn at the wide end of the triangular plates [4]. For our prototype, $l = 0.9144\text{m}$ and $W = 0.1026\text{m}$; therefore, $\Delta l = 1.43789 \times 10^{-3}\text{m}$. The frequency at which Δl is one-half wavelength is then given by

$$f_{\lambda/2} = \frac{c}{2\Delta l}, \quad (2.4)$$

where c is the speed of light. For our prototype, this frequency is 104.32 GHz, which is well beyond the desired operating frequency range of this TEM horn. However, this is not the only potential frequency limit associated with the TEM horn.

As long as the width-to-height ratio of the TEM horn is maintained throughout the entire length of the antenna, the antenna can be modeled approximately as a microstrip transmission line. Owing to this, we are then able to apply transmission line theory to accurately predict the upper and lower bandwidth limits of the TEM horn. This proves to be a tremendous advantage to the designer, since the only required data that must be known is the antenna's overall length, L , its width at the aperture, W , and total aperture height, $2h$.

To begin the process of determining the theoretical bandwidth limits of our prototype TEM horn, we return to the discussion of modes. As previously stated, the TEM mode is the only one desired in order to preserve the integrity of the incident fields. Thus, if any higher order transverse electric (TE) or transverse magnetic (TM) modes are excited, some distortion of the output waveform will occur. Fortunately, placement of conductors along the longitudinal axis of the antenna structure will prevent any induced currents from flowing across the width of the antenna structure (perpendicular to the antenna's boresight axis). This, alone, will serve to sufficiently deter the excitation of any TE modes within the TEM horn.

The same does not hold true for the TM modes, since there are no physical barriers, or mode filters, preventing their occurrence. From [3], only the TM_{02} mode, where the electric field has a component in the direction of propagation, has a significant probability of being excited, and thus, potentially causing distortion of the incident waveform. This mode can occur at any frequency higher than the limit governed by the equation

$$f_{TM_{02}} \approx \frac{\sqrt{10}c}{2h}, \quad (2.5)$$

where c is the speed of light, and $2h$ is the total aperture height. For our prototype TEM horn, this high frequency limitation caused by excitation of the TM_{02} mode will occur anywhere above 6.225 GHz. It is immediately evident that this is nearly a 40% reduction from the bandwidth we desire. Obviously, when we model our prototype TEM horn, this will be a frequency range to which we will need to pay close attention. If the TM_{02} mode is, in fact, excited at or near this frequency, then there will be a noticeable reduction in antenna efficiency (and possible distortion of the input signal) for all frequencies above this cutoff frequency, since power and signal transmission will now be passed through more than one mode.

Unfortunately, simply lowering the value of $2h$ in Equation 2.5 may not be the best idea. Certainly, this will increase the cutoff frequency for the TM_{02} mode; however, decreasing $2h$ will also serve to decrease the overall sensitivity of the TEM horn [4]. A reduction in the sensitivity for a receiving antenna will more quickly degrade its overall effectiveness than will excitation of the TM_{02} mode. Thus, there is a trade-off between maintaining a high sensitivity and a high TM_{02} cutoff frequency, with sensitivity being the more critical factor. For this reason, modifying the prototype TEM horn to increase the TM_{02} cutoff frequency would not be a wise decision.

On the low frequency end of our desired bandwidth, we find that the sensitivity is again the limiting factor. Specifically, it can be shown that under the assumptions given in [4], the low frequency limit can be expressed by

$$f_{L_{6dB}} \approx \frac{c}{4\pi L} , \quad (2.6)$$

where $f_{L_{6dB}}$ is the frequency where the sensitivity is 6dB less than its nominal (passband) value, L is the overall length of the antenna, and c is the speed of light. For the given values of the prototype TEM horn, $f_{L_{6dB}} = 13.054$ MHz. This would imply the prototype has a 3dB frequency of 26 MHz. These will be areas that will need to be viewed with interest when we analyze the prototype TEM horn using the software package described later. Although the above is a theoretical calculation, we should expect some degradation in performance beginning in the 15-20 MHz range.

D. LORENTZ RECIPROCITY THEOREM

One final element that will prove critical not so much in the design portion of this research, but rather in the analysis of the near fields that follows, is the Lorentz Reciprocity Theorem. One result of reciprocity is that the radiation pattern of a particular antenna is the same whether it is used as a transmitter or receiver [2]. This is an important fact to remember, since it is sometimes easier to analyze an antenna simply as a transmitter than a receiver. Due to reciprocity, then, the radiation pattern and behavior of the antenna in a transmitting mode will be no different (save for the direction of propagation) than the corresponding antenna acting as a pure receiver. As stated, this will apply more to when we begin to analyze actual antennas. It is provided now to emphasize that even as we begin our discussion of the impulse receiving antennas, any discussion of their field patterns would be the same, even if they were then operated as transmitting antennas.

III. ANALYSIS OF WIRE ANTENNAS AND SCATTERERS (*AWAS*)

As critical as the *NFLDVIS.M* function is to the visualization of the near electric and magnetic fields (near fields), so, too, is the *AWAS* program critical to the actual calculation of those fields. It is this, and *NFLDVIS.M*'s inherent dependence on *AWAS* output data, that makes the following discussion on *AWAS* necessary. While no substitute for the *AWAS User's Manual*, reference [5], this chapter should provide the user of *NFLDVIS.M* with enough detail and applicable background to create antenna geometries in *AWAS* and obtain valid output data. The information contained in this chapter is taken, for the most part, directly from the *AWAS User's Manual*. Included, though, is insight into *AWAS* as it applies to this thesis and to the user of *NFLDVIS.M*.

A. BACKGROUND

AWAS is a self-contained program which calculates the electric and magnetic fields (E- and H-fields) generated by a wire antenna or scattering structure in the near-field or far-field regions of space. As its name implies, *AWAS* is only able to calculate the fields for wire structures like the one shown in Figure 2. It generates the E- or H-fields from the currents that are induced along the wires. Wire antennas and scatterers are structures made of wire-like conductors, whose radii are much smaller than both their lengths and the wavelength of the operating frequency. Antennas which can easily be modeled by such wire structures are simple dipoles, V-antennas, wire loop antennas, tower broadcast antennas, Yagi-Uda antennas, log-periodic arrays, and antennas made by printed-circuit techniques. The wire structure can be placed either in free space or above a perfectly conducting ground plane. It can have concentrated or distributed resistive loads, and it can be driven at one or more ports in the transmitting mode or be excited by a plane wave in the receiving mode. *AWAS* is capable of analyzing these wire antennas operating in either mode individually or both modes simultaneously. [5]

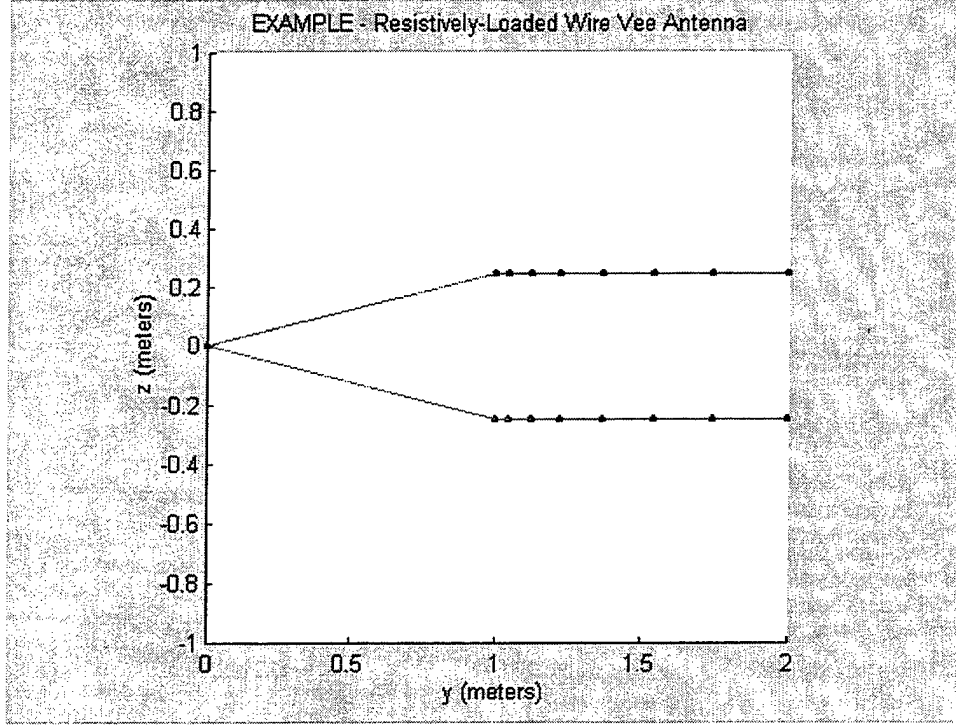


Figure 2. Example of Wire Antenna

AWAS consists of a windows-based user interface for data input and output and a powerful kernel for numerical analysis of the wire antennas. A typical computation cycle consists of data input, calculation of the electric and magnetic fields, and creation of the data output files. The program solves the following integro-differential equation for the current distribution along the wires, referred to as the two-potential equation,

$$\sum_{m=1}^N \int_0^{h_m} \left[\mathbf{u}_p \cdot \mathbf{u}_m I_m(s_m) g(r_a) + \frac{1}{k^2} \frac{dI_m(s_m)}{ds_m} \nabla g(r_a) \right] ds_m = \frac{\mathbf{u}_p \mathbf{E}_i}{j\omega\mu_0}, \quad (3.1)$$

where m is the index of the wire segment; p is the index of the wire segment for which the boundary condition is imposed ($p=1, \dots, N$); h_m is the length of the m -th wire segment; $I_m(s_m)$ is the current on the m -th segment along the local axis, s , of that segment; $g(r_a)$ is the thin-wire kernel derived from the free-space Green's function; r_a is the average distance between the observation point and the points along the wire circumference; $k = \omega\sqrt{\epsilon_o\mu_o}$; and \mathbf{u}_p & \mathbf{u}_m are unit vectors along the wire segment's axis. The equation is based on the boundary

conditions for the electric field on the wire axes. It is solved numerically using the Method of Moments, with a power series approximation for the current distribution and pulse basis functions. [5]

AWAS is available in three versions: *AWAS for DOS*, *AWAS for Windows*, and the *Extended DOS* versions. While *AWAS for Windows* is the most popular and better than the original DOS version, the *Extended DOS* version is by far the fastest and most robust of the three. Computation times using the *Extended DOS* version were found to be between 5 and 10 times faster than the *Windows* version, regardless of complexity. The following discussion of all files and data entry blocks, however, is transparent to the particular version used, and as such any version can use the files generated by the other. In the case of this thesis, construction of all input files using the *Windows* version and actual calculation of the near fields using the *Extended DOS* version was found to be the most user-friendly. As such, all steps for proper data input contained in the next section are based on the *AWAS for Windows* version.

B. DATA INPUT

To begin the input phase of the program, first execute the *AWAS* program as per your particular operating system. Once open, you should see the main *AWAS* window as displayed in Figure 3. Either click on the “INPUT” icon on the toolbar or select the “Input” menu from the menu bar at the top of the window.

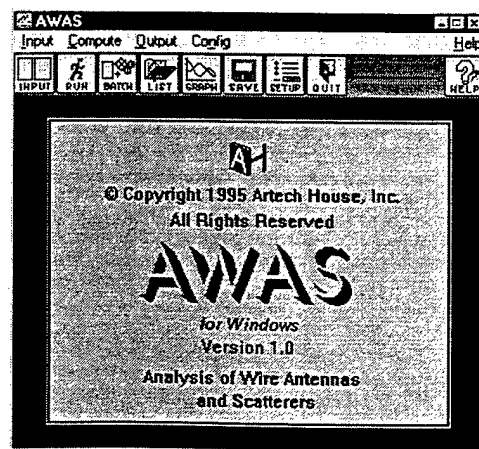


Figure 3. Main *AWAS* Window

The Input Data window should then appear in the lower portion of the screen, and the Show window should simultaneously open in the upper portion of the screen, as shown in Figure 4. The Input Data window is used to define all data necessary to create and specify the geometry of the wire structure to be analyzed. In addition, it is herein that the user will specify the frequency range and near-field group(s) over which the analysis will be conducted. The Show window provides a 3-dimensional (3-D) real-time view of the wire antenna as it is being constructed.

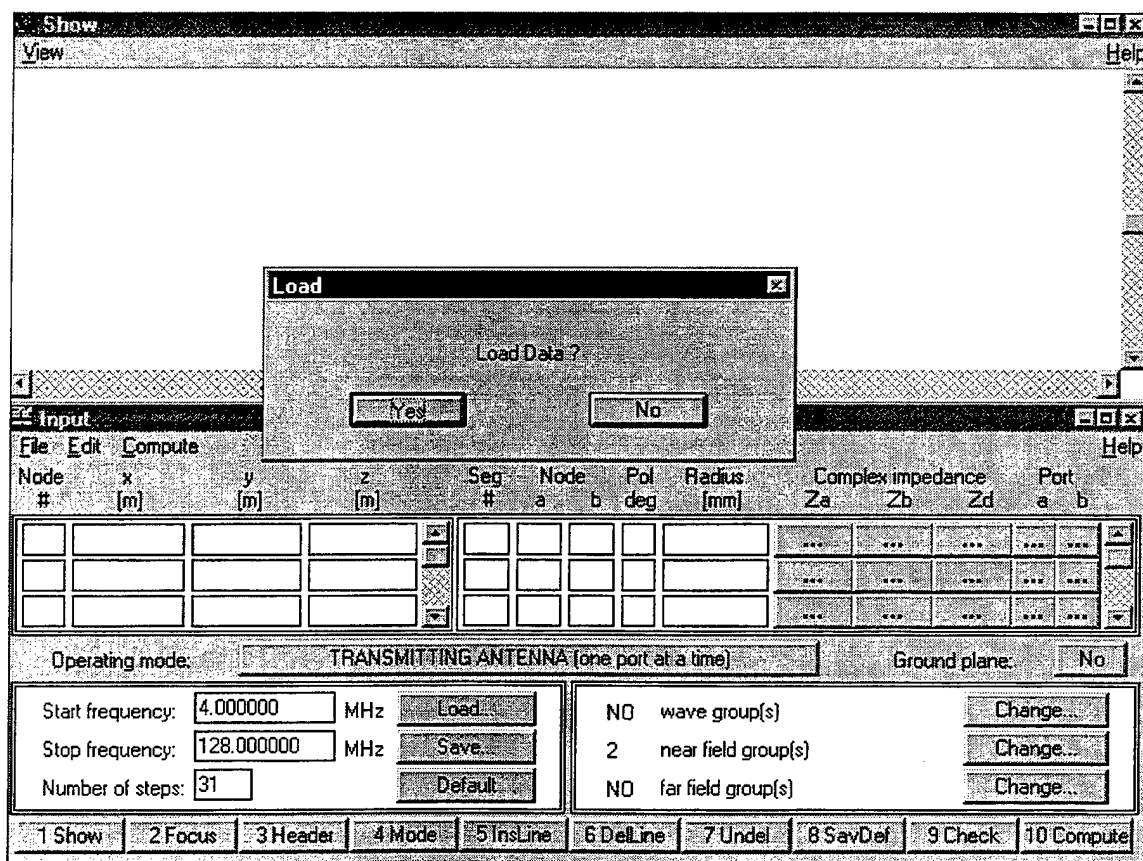


Figure 4. AWAS Input Data & Show Windows at Start-Up

To begin the input process, select "Yes" on the Load Input Data window to modify any input files that have already been created, or select "No" to create a new antenna geometry, its operating frequencies, and the location(s) of the associated near-field group(s).

All data input and modification is accomplished within the Input Data window through either direct entry (as for the nodes and segments) or through a series of pop-up windows (as for the plane wave groups, near-field groups, impedances, and ports). Also, all input data should be in SI units (meters, amps, etc).

In order to create the wire antenna structure, remember that it consists of a set of straight-line wire segments. Each segment connects two points in space (nodes), and each node is defined by its Cartesian coordinates (x, y, z). Figure 5 shows an example of how the input data is entered and assembled. The Show window displays a real-time sketch of the structure defined by the given nodes and segments.

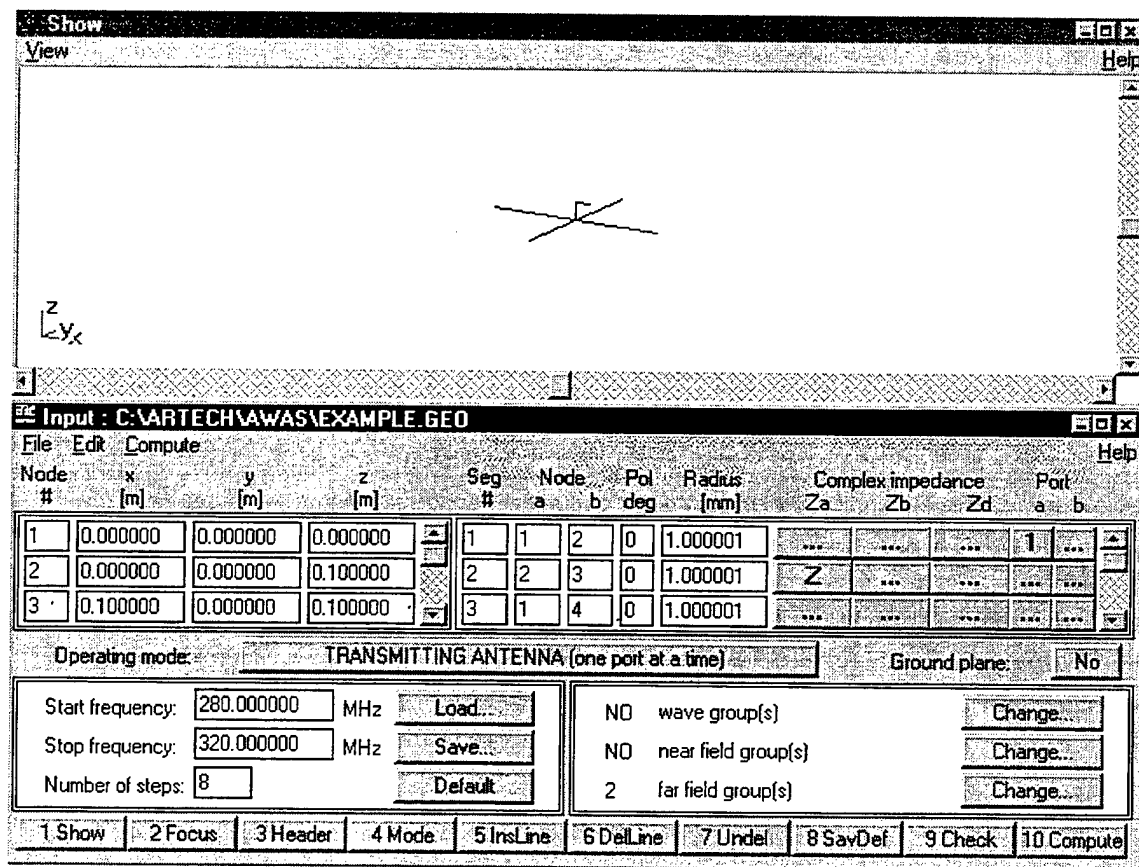


Figure 5. Example of AWAS Data Input & Antenna Display

Due to the specific relationship between the *AWAS* input and output files and *NFLDVIS.M*, certain restrictions have been placed on the construction of both the wire structure, itself, and the near-field groups for any analysis that will be carried over to *NFLDVIS.M*. First, the antenna must be oriented along the y -axis, i.e. the y -axis is the boresight axis of the antenna. Second, node 1 in the input *.GEO* file must be located at the source (feed) of the antenna, and the angle between the y -axis and a line drawn between nodes 1 and 3 must be equivalent to the angular dimension of the flared section of the TEM horn. Third, for all transmitting antennas, the input voltage at the feed must be +1 Volt to provide the “steady state” fields as output, and for all receiving antennas, the input plane wave should have an amplitude of 1 Volt/meter. These fields are then convolved with a Gaussian input impulse in *NFLDVIS.M* to provide the corresponding near fields due to the input impulse. Fourth, there must be an odd number of points (or none at all) along each axis of a near-field group. Fifth, all near-field groups must be symmetric about the y -axis and lie within either the xy -plane or the yz -plane. And lastly, no point within a near-field group may touch the antenna feed point or, to the greatest extent possible, any wires on the antenna structure. Because all near-field groups in *AWAS* must be rectangular in shape, and in order to see the near fields within the flared section of the TEM horn, some points in a near field group created in the yz -plane will have to either lie on or outside of the wire structure. This will not create any problems as long as the restriction on nodes 1 and 3 (as discussed above) is met, since *NFLDVIS.M* will then be able to “filter” these data points out of the final near field calculations. Further data input limitations due to *AWAS*, itself, are discussed later in Section 3.D.

Once the structure is complete, select “Compute” at the bottom of the window, and *AWAS* will check all nodes, segments, frequency steps, etc., for proper values and ranges. After checking the continuity of the wire antenna structure, *AWAS* then saves all input data, and calculates the near fields.

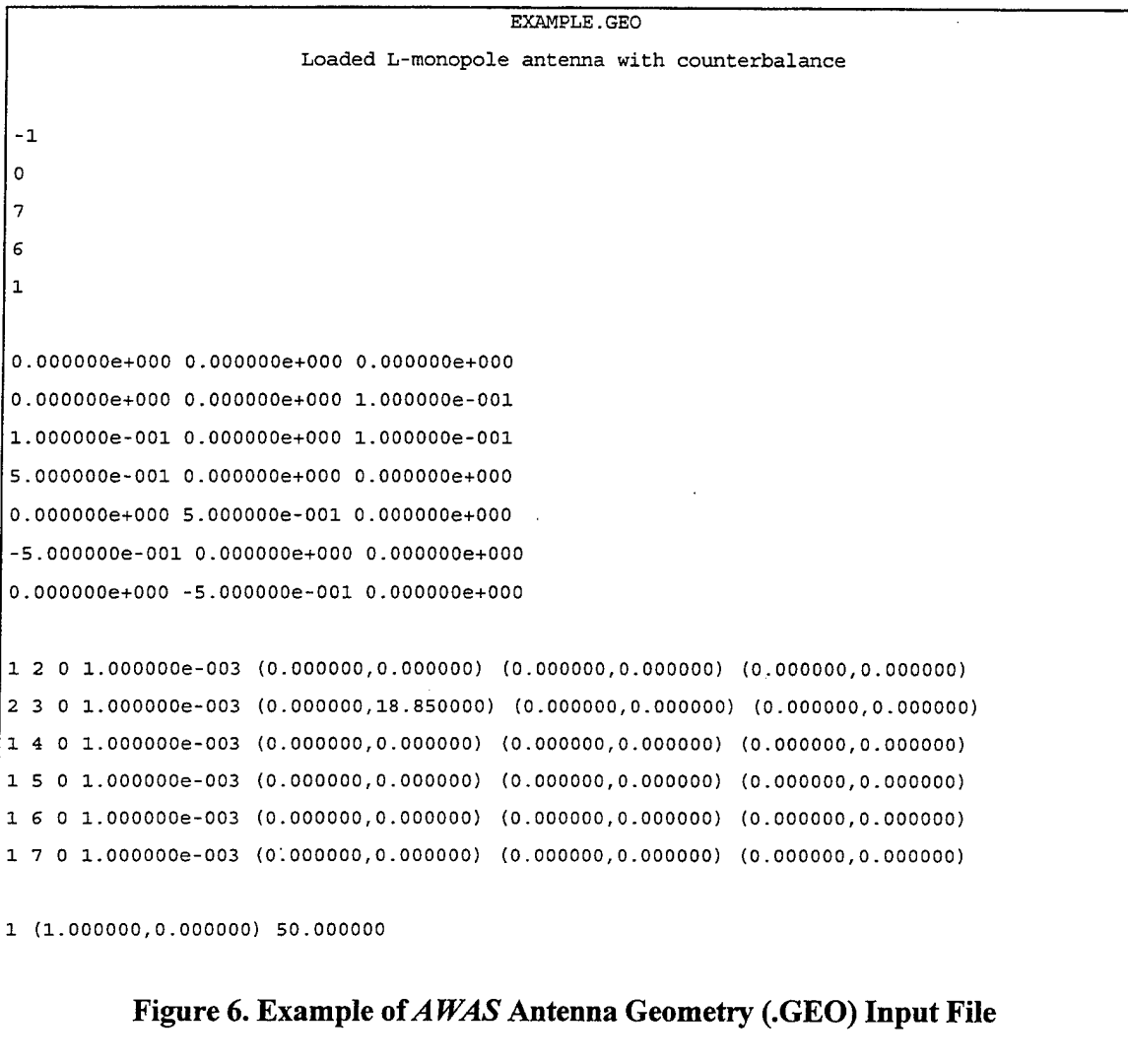
C. INPUT AND OUTPUT FILES

All input data is saved within five separate files whose name extensions are *GEO*, *FRQ*, *PWE*, *NFP*, and *FFP*. The antenna geometry data (node coordinates, segments, etc) are saved in files with the default *.GEO* extension. An example of one is shown in Figure 6. The first three lines of the file are the header, which can contain any description the user supplies. Each line is 76 characters long, including blanks. The entry in the fourth line indicates the operating mode selected for the particular structure. In this example, the "-1" indicates it is operating as a transmitter only, and only one port (feed) at a time is excited. A "+1" indicates a transmitting antenna with all ports excited, and a "2" would indicate a receiving antenna. The fifth line indicates the presence of a ground plane with a "1". A "0" in the fifth line indicates the antenna is in free space. Following this, line six contains the number of nodes; line seven is the number of wire segments; and line eight is the number of ports (feeds).

The next major group of entries defines the location of each node in Cartesian coordinates (x,y,z). In this example, the first node is located at point (0,0,0), the origin. All distances here are in meters, as is the same in the Data Input window.

The next major group describes the location of each segment. The first two characters in each line represent the start and termination nodes, respectively, of the segment. The third character is the polynomial degree chosen for calculating the two-potential equation, and the fourth number is the radius of the wire segment (in meters). The final three complex number sets indicate the concentrated impedance at the start node (Z_a), the concentrated impedance at the termination node (Z_b), and the impedance uniformly distributed along the entire segment (Z_d), respectively.

The final group corresponds to the ports on the antenna. Each line represents the segment number, complex driving voltage, and nominal impedance of each particular port. If the port is located at the origin of the segment (as is the case for the example shown), then the first integer will be positive. If the terminal end of segment #1 were driven for this example, then the first integer would be a "-1".



The structure of the *.GEO* file is critical to *NFLDVIS.M*, as the function reads and extracts the pertinent data necessary for reconstructing and visualizing the near fields using MATLAB and its associated graphic tools.

The *.FRQ* file extension indicates those files which contain the data defining every frequency for which *AWAS* will analyze the antenna structure. This file exists solely to pass the frequency data to the *AWAS* program kernel and is not used by the *NFLDVIS.M* function. Likewise, the *.PWE*, *.NFP*, and *.FFP* files contain the data which define the incident plane wave(s), the near field zone(s) of interest, or the far field zone(s) of interest to the user. They, too, are used strictly by the *AWAS* program kernel.

The results of the *AWAS* analysis are written to seven separate default output files. At the user's discretion, these files are then also saved under a user-defined name with seven different file extensions: *OUT*, *SPM*, *YPM*, *ZPM*, *CUR*, *NFL*, and *FFL*. The only file extension that will be discussed is the *.NFL* file, since this is the one which contains all the near E- and H-field data for the desired frequency range, and thus, is the primary source file for *NFLDVIS.M* for visualization of the near fields. An example is shown in Figure 7.

It should be noted that the *.NFL* file does not contain a header as its first line. This was inserted using a text editor exclusively for use within this discussion. The actual first line of every *.NFL* file begins with an integer representing the number of near field groups analyzed by *AWAS* (2 for this example).

Following this line, the second line contains the start and stop frequencies, f_1 and f_2 , respectively, followed by the number of frequency steps, N_f . It should be noted that the frequency step size, df , should always be set equal to the starting frequency, f_1 , and the number of frequency steps must be at least 16, if the user wishes to view the fields in the time domain. For the example, the structure was analyzed over the frequency range of 5MHz through 30MHz in 5MHz steps (5 steps total).

The third line indicates the antenna type in the same manner as indicated in the *.GEO* file. Our example here represents a transmitting antenna exciting one port at a time. The fourth line indicates the number of ports present on the antenna structure (1 for this example), and the next line is the number of plane-wave groups incident upon the structure (0 for our pure transmitter, as common sense would dictate).

The next block of data is the first array of near-field values. All of the following blocks will look similar to this one. This first block of near-field data applies to the first near-field group and represents the fields calculated at frequency f_1 . The next block would represent the data for the second near-field group at frequency f_1 , and so on until the near fields are calculated for every near-field group at this first frequency. After that, the data blocks cycle through all the near-field groups again at each successive frequency.

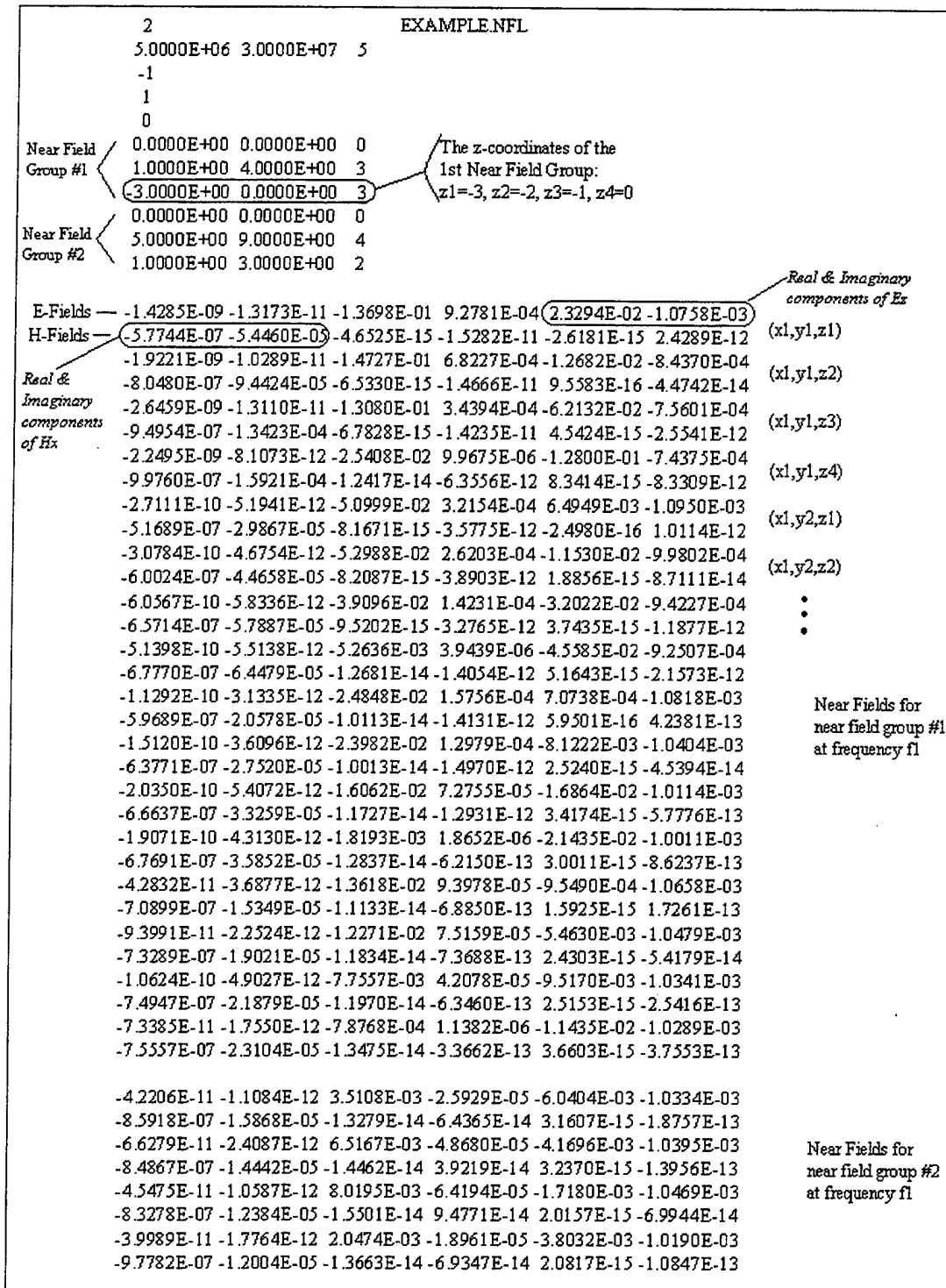


Figure 7. Example of AWAS Near-Field (.NFL) Output File

Within each data block, the first two lines represent the E- and H-fields, respectively, for the first planar (spatial) coordinate (x_1, y_1, z_1) . Lines 3 and 4 of that block represent the E- and H-fields for the spatial coordinate (x_1, y_1, z_2) , and so on through each spatial coordinate of that near field group. It should be evident that the data cycles through the z -coordinates, then the y -coordinates, and finally the x -coordinates, in that specific order.

Looking across the first line of near field data, the E-field is presented as six floating-point numbers. These represent the real and imaginary components of the E-field in the x -direction (E_x), the real and imaginary components of the E-field in the y -direction (E_y), and those of the E-field in the z -direction (E_z), respectively. Similarly, the second line contains the H-field components along the three Cartesian axes, H_x , H_y , and H_z . It should then be evident that for every block of near field data, all odd rows contain the E-field data and the even rows contain the H-field data. This specific structure is critical to the function *NFLDVIS.M*'s ability to access and manipulate all the near field data for graphical analysis. Consistent here, too, is the fact that all quantities are given in SI units (meters, Hz, volts/meter, amps/meter, etc).

For the case of the receiving antenna analysis, the *NFL* output file differs only slightly from its counterpart for the transmitting antenna case, Figure 7. As noted in Figure 8, line 3 will have a "2" representing the receiving antenna, while line 5 indicates the presence of a singular incident plane wave. In addition, prior to the block(s) of data representing each near-field group, an extra block of data is inserted which describes the incident plane wave information, as shown.

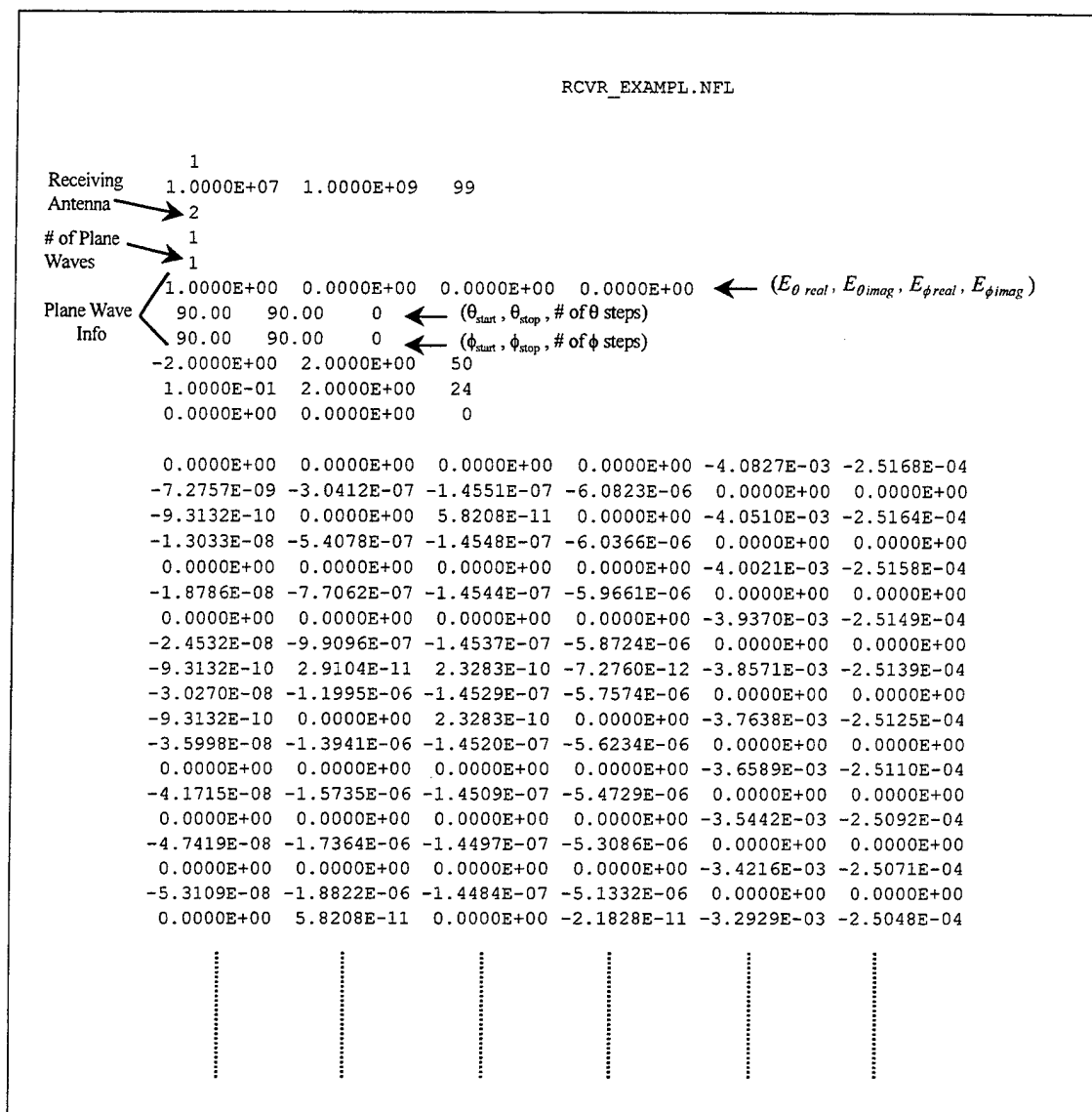


Figure 8. Example of *AWAS* Near-Field (.NFL) Output File

D. PROGRAMMING NOTES

Table 1 lists the upper limits of the various input data for each of the three *AWAS* versions. It should be noted that besides being slower than the *Extended DOS* version,

AWAS for Windows only allows a maximum of 199 frequency steps, a limit that severely handcuffs the UWB antenna designer.

Table 1. Upper Limits for All <i>AWAS</i> Versions			
<i>Item</i>	<i>AWAS for Windows</i>	<i>Extended DOS</i>	<i>AWAS for DOS</i>
<i>Max # of Nodes</i>	<i>500</i>	<i>999</i>	<i>50</i>
<i>Max # of Segments</i>	<i>320</i>	<i>999</i>	<i>50</i>
<i>Max # of Ports</i>	<i>16</i>	<i>99</i>	<i>9</i>
<i>Max # of Freq Steps</i>	<i>199</i>	<i>999</i>	<i>999</i>
<i>Max # of Plane Wave Groups</i>	<i>8</i>	<i>9</i>	<i>9</i>
<i>Max # of Near Field Groups</i>	<i>9</i>	<i>9</i>	<i>6</i>
<i>Max # of Far Field Groups</i>	<i>9</i>	<i>9</i>	<i>9</i>
<i>Max # of Coordinate Steps per Group</i>	<i>199</i>	<i>999</i>	<i>99</i>
<i>Max Polynomial Degree</i>	<i>9</i>	<i>9</i>	<i>9</i>
<i>Max # of Unknowns</i>	<i>640</i>	<i>2000</i>	<i>150</i>

As stated in the *AWAS User's Manual*, all input data supplied to *AWAS* is further restricted to certain ranges in addition to the above maximum limits. These restrictions on the input data are as follows:

- The number of nodes must be at least 2.
- The number of segments must be at least 1.
- The number of ports must be between 1 and 16 for an antenna.
- The z-coordinate for a node must be ≥ 0 if the ground plane is present.
- The node indices for a segment must be between 1 and the total number of nodes.

- The segment radius must be positive.
- The polynomial degree must be 0 for automatic segmentation or between 0 and 9 for manual segmentation.
- The real part of a loading impedance must be ≥ 0 . The imaginary part can be positive, negative, or zero.
- The segment index for a port must be between 1 and the total number of segments.
- At least one port's electromotive force must be > 0 for a transmitting antenna.
- The port nominal impedance must be > 0 .
- The start and stop frequencies must be positive.
- The number of frequency steps must be at least 1.
- The number of plane wave groups must be between 1 and 8 for receiving antennas.
- The angle θ for the plane waves must lie between -180° and 180° (or -90° and 90° if the ground plane is present).
- The angle ϕ for the plane waves must lie between -180° and 180° .
- The number of near field groups must be at least 1.
- The upper and lower z-coordinate limits in the near field groups must be ≥ 0 if the ground plane is present.
- The number of coordinate steps in the near field groups must be ≥ 0 .

The information presented within this chapter is by no means meant as a substitution for the *AWAS User's Manual*; however, it should give the antenna designer enough information to quickly create, load, and compute the near E- and H-fields for a particular antenna structure. Once those near fields are calculated, though, it is the function *NFLDVIS.M* which takes that data and transforms it into a visual display for complete graphical analysis of the antenna's effectiveness at any given frequency.

IV. VISUALIZATION OF A PROPAGATING IMPULSE

In order to visualize the transient near-fields produced by an UWB antenna, and more specifically, by the TEM horn, a graphical user interface (GUI) was developed using the MATLAB programming language as its base. The GUI and the entire algorithm developed to conduct the necessary matrix calculations were written as the function *NFLDVIS.M*, included here as Appendix A.

A. SOFTWARE OVERVIEW

NFLDVIS.M, heretofore referred to as “the software,” was developed as the key analysis tool for the UWB antenna designer. Its system requirements are a 486DX or faster processor with Windows 95 or Windows NT as the operating system, an 8-bit graphics adapter, and 16 MB of Random Access Memory (all MATLAB 5.x constraints). The actual processor speed, amount of Random Access Memory (RAM), and graphics requirements, though, will be dictated by the size of the near-field arrays calculated within *AWAS*. The system used in the development and testing of the software was a 200MHz Pentium Pro with 132MB of RAM. This proved adequate for manipulation of arrays as large as $101 \times 101 \times 1024$, where the first two dimensions represent the physical size of the near-field array in the xy - or yz -plane, and the third dimension is the number of time steps (directly derived as twice the number of frequencies analyzed).

The software must be run under MATLAB 5.0 or higher, due to the extensive use of 3-dimensional (3-D) matrices and also the integration of new graphics and user-interface controls unavailable in previous MATLAB versions. Although a complete User's Manual is not included, the highlights as they apply to this thesis will be presented and reinforced throughout the main body of this document.

The software was designed to specifically work with TEM horns or wire V-antennas which were constructed and tested using *AWAS* and physically oriented along the y -axis in 3-D space. Due to this physical orientation, only the y - and z -components of the electric

field and the x -component of the magnetic field are ever made available for full analysis and visualization (E_z and H_x are the major near-field components, while any values of E_y represent noise in the direction of propagation). While this may at first appear to be the most simple or trivial scenario for an antenna alignment, it should be noted that all analysis here can be easily manipulated for application to any direction of propagation through the use of “direction cosines”. Simply put, what you actually visualize here is a true representation of the near fields along the antenna’s boresight axis.

The software does, however, present certain limitations to the designer. As mentioned in Chapter III, *AWAS* will only construct rectangular near-field grids. This presents a problem any time the designer wishes to view the near fields in the yz -plane, since the TEM horn is not rectangular throughout its entire shape. In order, then, to filter out the points on or outside of the antenna structure (especially near the feed), the software calculates the half-angle dimension of the TEM horn, then simply replaces those values outside the designated angle with zeroes. To do this, it is assumed that the angle between the y -axis and a line drawn through the 1st and 3rd nodes in the geometry (*.GEO*) file accurately duplicates the angular dimension of the flared portion of the TEM horn. This limitation should pose little in the way of additional effort on the part of the designer, but this knowledge is essential prior to construction of any antenna geometry in *AWAS*. This limitation does not apply for the case of a near-field array lying in the xy -plane due to the fact the TEM horn has no physical sides, and thus, field points within the rectangular array will not lie on or near any physical part of the antenna.

Throughout the course of this chapter, the terms “near-field groups” and “near-field arrays” will be used. As a matter of clarification, “near-field groups” are the physical xy - or yz -plane limits imposed by the user within which *AWAS* will calculate the near E- and H-fields. “Near-field arrays,” or “near fields,” are the matrices which contain the field strength values for each point within a near-field group and over the entire range of frequencies. Whereas a near-field group is a 2-D planar matrix containing x - and y -coordinates or y - and z -coordinates in space, a near-field array is a 3-D matrix containing the E- or H-field

magnitude that *AWAS* calculated for each point within the near-field group and over the entire user-defined frequency range for which the antenna was tested. Although this fact is not critical to the user of the software, its knowledge will aid in understanding how the software is able to create the graphical analysis that is fundamental to the design process.

B. INITIALIZING THE SOFTWARE

After opening MATLAB to the correct directory and typing "nflavis" at the command line prompt, the entire screen will fill with a figure window containing the GUI. The GUI's purpose is to provide a media through which the antenna designer/software user may conduct all analysis of an antenna, as well as plotting, printing 3-D figures, and saving files entirely from within the GUI window and with only the use of a computer mouse. The GUI consists of two sets of axes for plotting both the data and an optional colorbar, the user interface controls (uicontrols) such as the push-buttons and pop-up menus, and the user interface menus (uimenu) such as the "Plot" and "Print" pull-down menus, as depicted in Figure 9. It should be readily apparent that some uicontrols have darker text than others. Within uicontrols and uimenu, the text is dark when the uicontrol/uimenu is activated, and the user may press or use it. Light or nearly transparent text indicates the uicontrol/uimenu is deactivated, and it will not function, regardless of the user's attempts. In the case of the sliders at the bottom of the window, the sliding button disappears when these are deactivated.

Immediately upon initializing the software, the user is provided with four options through which to proceed:

1. Select "Start," and the user may start the analysis of a particular antenna;
2. Press the "Demo" pushbutton, and a complete demonstration (demo) of the software will run;
3. Select "Movie," and either a new movie can be created using the data from the last analysis, or a previously-created movie may be replayed, or;

4. Use the "Plot" uimenu to plot the frequency-domain fields of the last antenna analyzed.

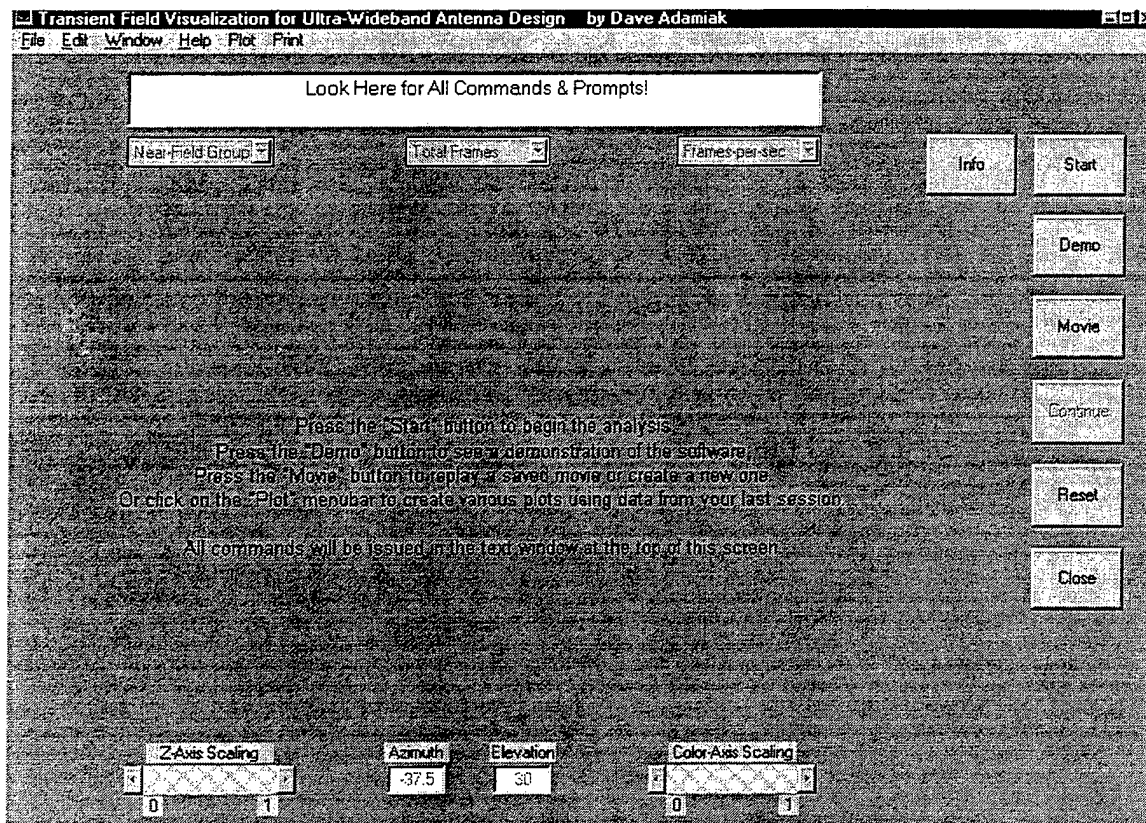


Figure 9. Initial Window on Start-Up of NFDVIS.M

The "Reset" and "Close" pushbuttons are available at all times, and while the function of the "Close" button should be obvious, the "Reset" button's function is exclusively to restore all uicontrols to their default values while simultaneously returning the user to the screen shown in Figure 9. The new user of *NFDVIS.M* should first select the "Info" button and then run the demo before beginning the analysis of a particular antenna in order to acquaint him/herself with the software, its capabilities, and the normal flow of the data and graphics subroutines.

The "Info" button and its associated subroutine specifically serve to introduce the new user to the software and its capabilities and limitations. It is merely a quick synopsis

and by no means a substitution for the User's Manual, reference [6], but it does provide the user with the requisite knowledge to get started on the right track.

Pressing the "Demo" button activates the demo package. This is provided to illustrate the inherent capabilities of the software and familiarize the new user with the general flow of analysis, data manipulation, and graphics presentation. It does not, however, actually conduct any calculations of the fields; rather, it simulates such data analysis to ensure the demo presents the entire graphical-analysis package without being too time-consuming. It is constructed in a user-friendly manner that waits for cues from the user prior to moving forward to the next step. The entire demo may be run in a matter of a few minutes, depending solely upon the delays desired by the user. The general flow of the demo consists of:

1. File selection by the user;
2. Simulated near-field array construction;
3. Frequency-domain plotting of the E-field;
4. Application of the gaussian impulse;
5. Frequency-domain plotting of the E-field modified by the input impulse;
6. Transformation of the frequency-domain E-field to the time domain, and;
7. Creation of a movie representing the time-domain representation of the E-field, e.g., the propagation of the gaussian impulse through the antenna array.

The specific methodology used to create the data arrays is contained in Section 4.A.3, while the entire upper-level graphics package is covered in detail in Sections 4.A.5 and 4.A.6. For a complete step-by-step discussion of the demo package, refer to reference [6].

C. DATA INPUT & CALCULATIONS

The actual data input and calculations begin by selecting the "Start" button. Immediately, the user is tasked with selecting the appropriate *AWAS* .GEO and .NFL files for the desired antenna, as shown in Figure 10. It is critical that both files should correspond to the exact same antenna. Normally, this means the two files will share the same name and differ only by their file extensions (e.g., *TEM1.GEO* and *TEM1.NFL*), since *AWAS* saves all input files in this manner and prompts the user to do likewise with the output files. In addition, both files must reside in the same Windows/DOS sub-directory.

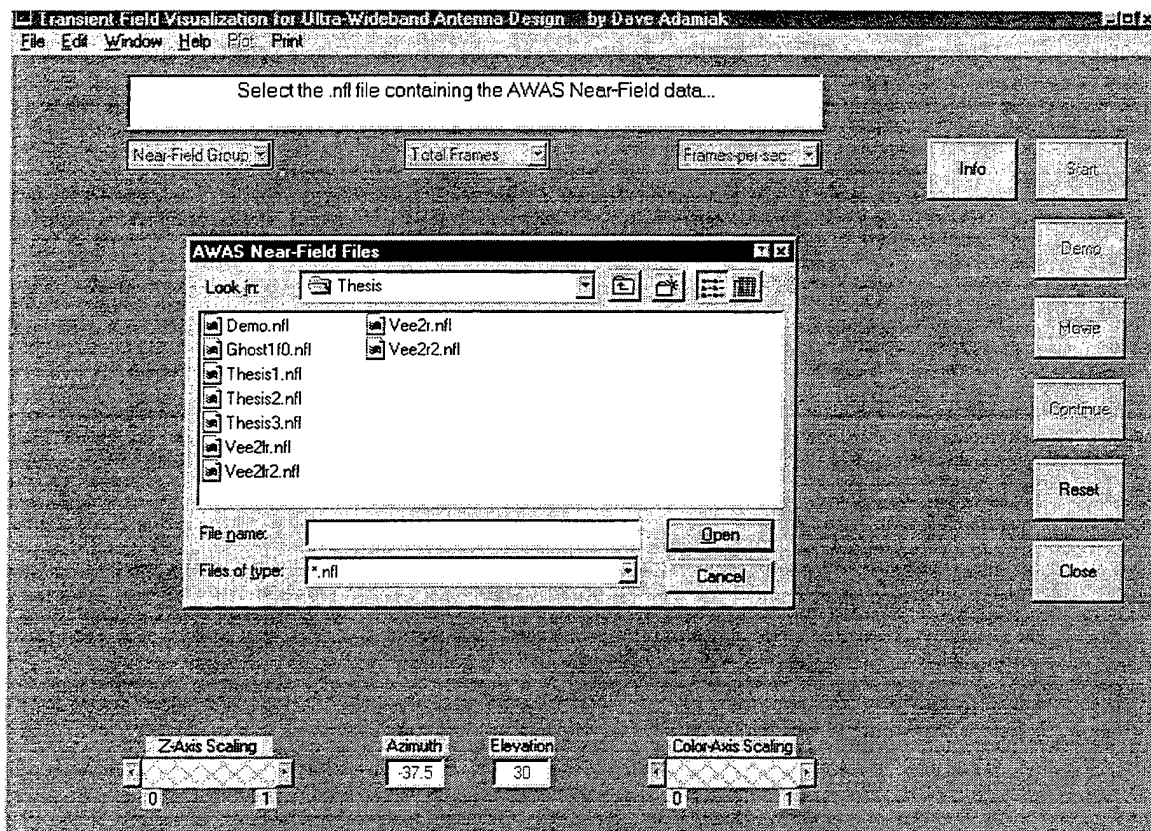


Figure 10. Selection of .NFL File for Analysis

Following selection of the appropriate files, the software extracts the necessary physical dimensions of the antenna and the near-field groups, and then asks the user to determine which near-field group to analyze and plot. It then reads the near-field data from

the *.NFL* file and creates separate 3-D matrices in MATLAB containing the field strength of the y - and z -components of the E-field and the x -component of the H-field in the frequency domain ($E_y(f)$, $E_z(f)$, and $H_x(f)$, respectively). For the case of a near-field group constructed in the xy -plane, the first dimension of the 3-D array will be the number of points in the x -direction, the second dimension will then equal the number of points selected along the y -direction, and the third dimension will be the number of frequencies tested. The size of the final array constructed for the yz -plane near-field group, likewise, will be determined by the number of points in the y - and z -directions and the number of frequencies tested. However, if the ground plane was present in the *AWAS* calculations, the software compensates for this in the construction of the matrices and creates the necessary data for the near-field array as if the ground plane were not present, i.e., it constructs a mirror image of the *AWAS*-calculated array and merges the two together along the y -axis. Regardless of the type of near-field group, though, the 3-D near-field arrays will always be symmetrical about and include the y -axis.

After the frequency-domain near-field arrays are assembled in MATLAB, the user is prompted to provide identifying information that the software will attach to the title of every plot or movie generated from the computed near-field arrays. The software then creates a Gaussian impulse input wave, Figure 11, whose maximum value in the time-domain is 1 Volt. The pulsewidth is a function of the minimum frequency and the total number of frequency steps. For this example, the pulsewidth as measured at the 3dB points is 1.0 nsec. This pulse is then convolved with the frequency-domain fields by taking the Fast Fourier Transform (FFT) of the time-domain Gaussian pulse, Figure 12, and then multiplying this result with each near-field component. As Figure 12 illustrates, the frequency-domain representation of the Gaussian pulse is similar to that of a low-pass filter, albeit a very broadband one [7]. The net results, as shown in Figure 13, are near-field arrays in the frequency domain that conform to the shape of Figure 12. These “filtered” frequency-domain E- and H-fields will, through use of the Inverse Fast Fourier Transform (IFFT), convert easily to an accurate time-domain representation of the near fields within and in

close proximity to the antenna that result from the application of a Gaussian impulse voltage at the antenna port.

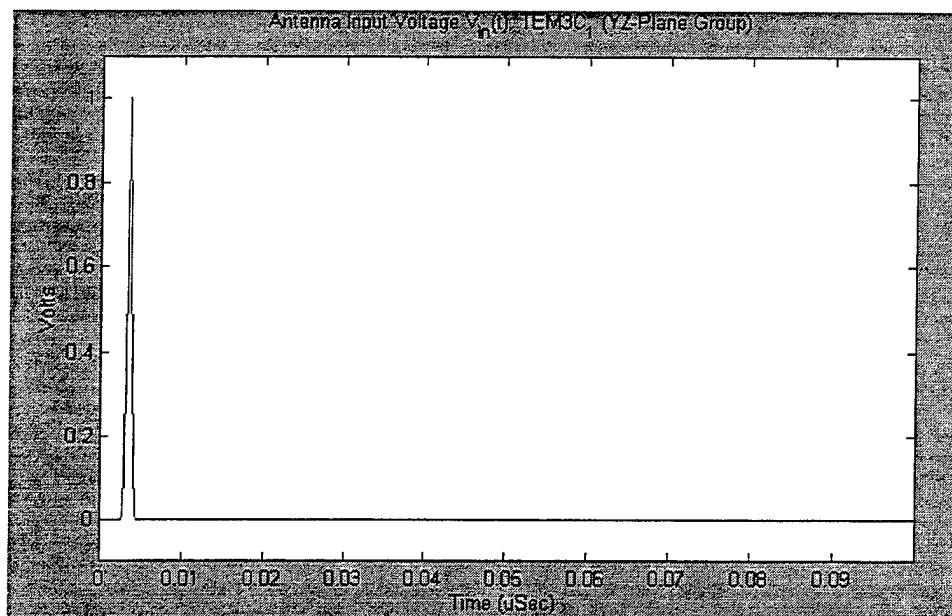


Figure 11. Gaussian Impulse Input Voltage

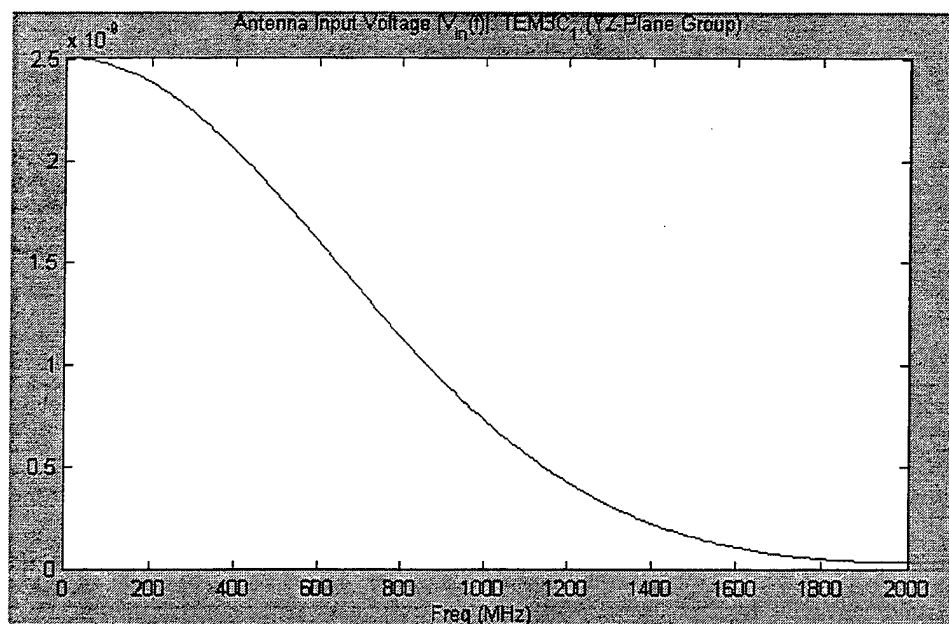


Figure 12. $|V_{in}(f)|$, The Frequency Spectrum of $V_{in}(t)$

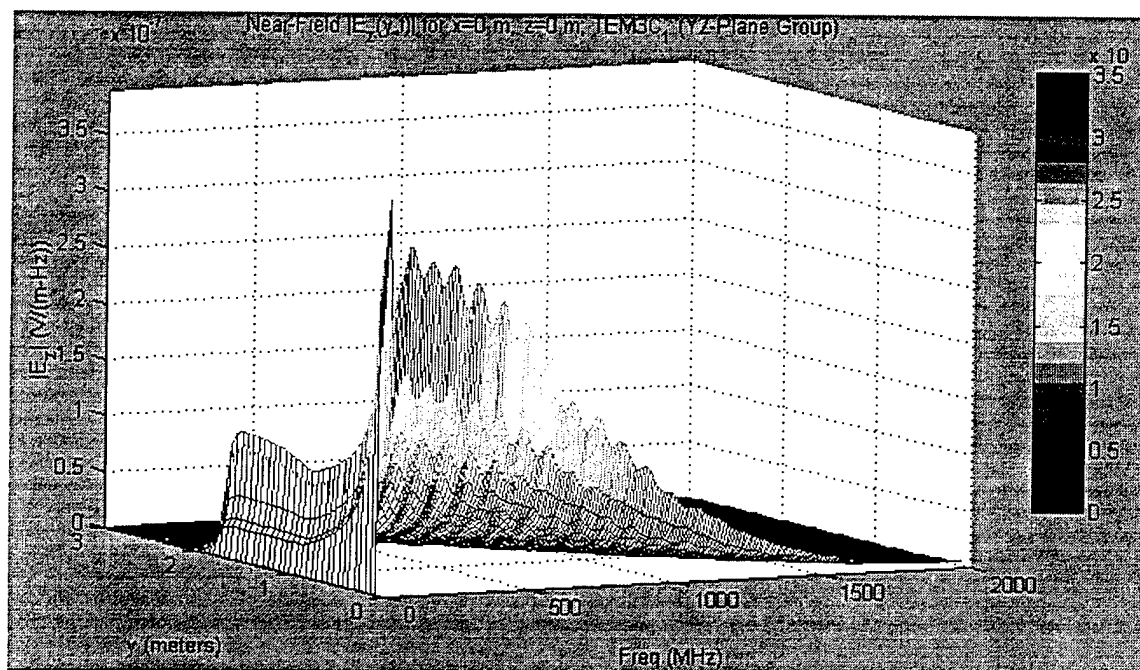


Figure 13. Example of "Filtered" Frequency-Domain E-Field

After the IFFT is applied along the 3rd dimension of the 3-D arrays, and only if the selected near-field group was in the yz -plane, the software will further filter the near-field arrays to eliminate the field values for points next to, on top of, or outside of the wires representing the flared portion of the TEM antenna structure. This last action completes the string of calculations and matrix manipulations necessary to create the variables that will be used for the various plotting and movie subroutines. The user is informed of the fact that all calculations are complete and then prompted to continue the analysis by either plotting the frequency-domain fields and the Gaussian input voltage or creating a movie of the near-fields in the time domain. The user may also elect to end the analysis there temporarily and return to complete his work at a later date, since all the variables needed to execute the plotting and movie functions were already saved automatically during the "Start" subroutine.

D. OUTPUT FILES

Rather than hold potentially large variables for the entire process of constructing the numerous near-field arrays, the software simultaneously saves all the critical variables within default *.MAT* files in the current directory and then deletes the variables from the workspace in order to free more memory for active matrix calculations. By saving the variables as *.MAT* files, they can then be readily accessed at any time in the future for plotting or movie construction. The eleven separate *.MAT* files are *Eyf.mat*, *Ezf.mat*, *Hxf.mat*, *Eyff.mat*, *Ezff.mat*, *Hxff.mat*, *Hdr.mat*, *Eyt.mat*, *Ezt.mat*, *Hxt.mat*, and *Movihdr.mat*. The first three contain the frequency-domain near-field arrays for each of the respective near-field components and are saved once the arrays are constructed from the *AWAS.NFL* data file. The second set of three files contain their respective "filtered" near-field arrays (the frequency-domain fields due to the input Gaussian impulse), and *Hdr.mat* contains a number of minor variables used to either support plotting of the frequency-domain fields or of the Gaussian impulse. These files are saved prior to transforming the arrays from the frequency-domain to the time-domain via the IFFT. The three time-domain near-field arrays are saved within *Eyt.mat*, *Ezt.mat*, and *Hxt.mat*, and *Movihdr.mat* contains the minor variables required to support construction of a movie. They are the last files saved prior to the "Start" subroutine concluding.

The above *.MAT* files form the data sets that both the "Plot" and "Movie" subroutines use for graphics display. Since these files can easily exceed 5MB in size, it is much easier to load and clear them as needed, as opposed to creating them through the "Start" subroutine each time a plot must be rendered. Also, the *.MAT* files will hold the near-field arrays and the associated minor variables until either the user deletes them manually or another antenna is analyzed. Because of this, the user can then conduct the analysis at his or her own pace. Also, as will be seen in the next two sections, the user may manually rename any or all of the aforementioned *.MAT* files using the operating system's file manager protocol, and thereby ensure the data never requires recalculation. This makes things easy when comparison of two antenna models is required/desired.

E. GRAPHIC DISPLAYS

Every plot of the frequency-domain near-fields or the input Gaussian function can be found beneath the “Plot” uimenu on the GUI’s menubar, as seen at the top left corner of Figure 9. All of the plots are grouped within the three sub-menus “Frequency-Domain Fields”, “ V_{in} ”, and “Filtered Freq Domain Fields”. Immediately after completion of the “Start” subroutine or anytime there exists data within either the default output files or other user-defined files as described in Section 4.A.4, the user may access and display the data merely by moving the mouse-pointer atop the “Plot” uimenu, clicking the left mouse button once, moving the mouse-pointer to the desired plot’s sub-menu, and then clicking the left mouse button again, as illustrated in Figure 14.

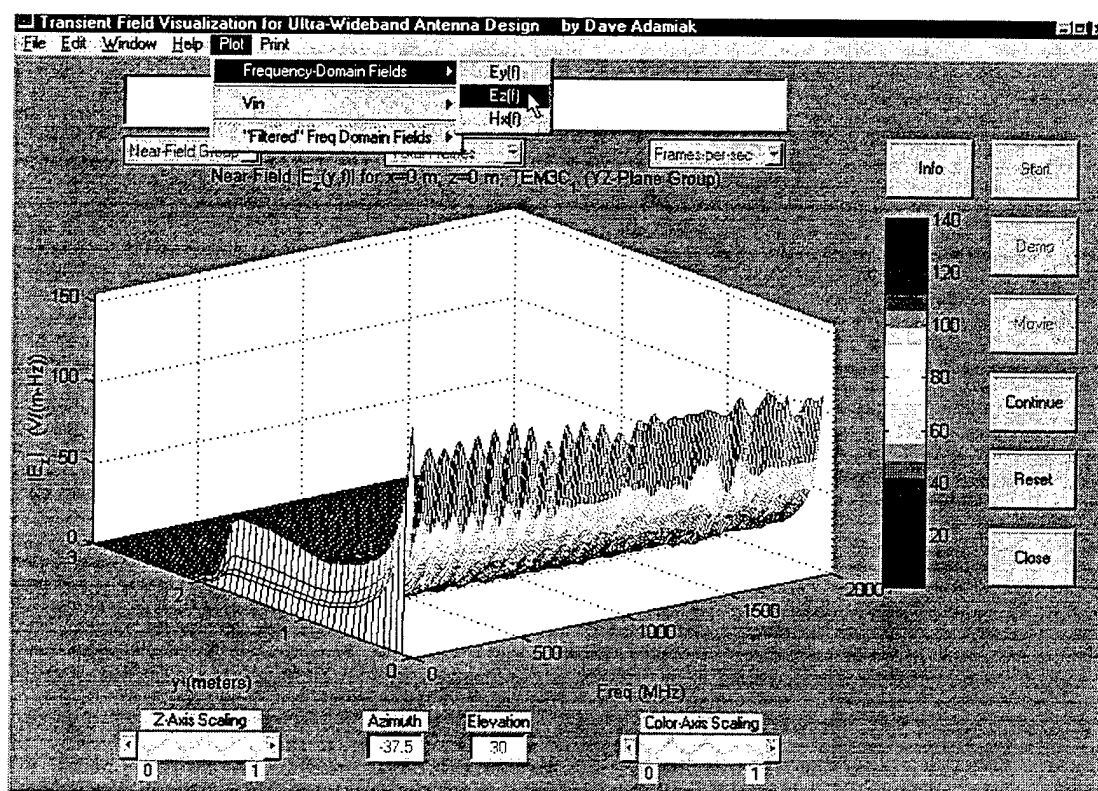


Figure 14. Example of E-Field Frequency-Domain Plot

In the example shown, a frequency-domain plot of the E-field as a function of both frequency and the y-axis position within the near-field group is rendered. Specifically, this is a plot of the magnitude of the E-field along the boresight axis of the antenna and over the

entire frequency range tested. The plot's colors range from blue at $E_z=0$ (cold) to red at $E_z=\max(E_z)$ (hot), and the exact color range and associated numerical values are duplicated in the colorbar to the right of the 3-D plot. If the user desires a different perspective angle to view the plot, he need only to place the mouse-pointer on top of the plot, press and hold the left mouse button, then move the mouse in any direction desired. As shown in Figure 15, a 3-D frame will first appear enclosing the original plot, and then it will move and resize according to the movements of the mouse by the user. Rotation is completely unlimited in all three dimensions; therefore, the user is able to view the 3-D plot from any perspective. This frame is merely a wire rendering of how the plot box will be reshaped once the user releases the left mouse button. Although not displayed in Figure 15, a clear text overlay will appear at the bottom left-hand corner of the GUI and provide real-time azimuth and elevation data regarding the view angle of the plot as the mouse and frame move. The azimuth and elevation text boxes at the bottom center of the window, however, are used within the movie subroutine only and will not provide any information to the user during any of the plot subroutines.

All of the plot subroutines within either of the two frequency-domain sub-menus will generate their plots in the same manner as depicted in Figure 14 and 15. The only difference will be the type of data used to render the 3-D graphic. Also, every plot will have the user-provided identifying information appended to the title. In the example shown, the user wished to identify this plot as that for the yz-plane near-field group from an antenna called "TEM3C₁".

The user may also elect to view the input Gaussian impulse voltage function in either the time- or frequency-domains. Selection of either graphic is no different than before, but this time, the rendered plot is only 2-D and cannot be rotated. An example is shown in Figure 16. The pulse width and total length of V_{in} , the input voltage pulse, are determined by the maximum frequency value and the total number of frequencies defined by the user in *AWAS*.

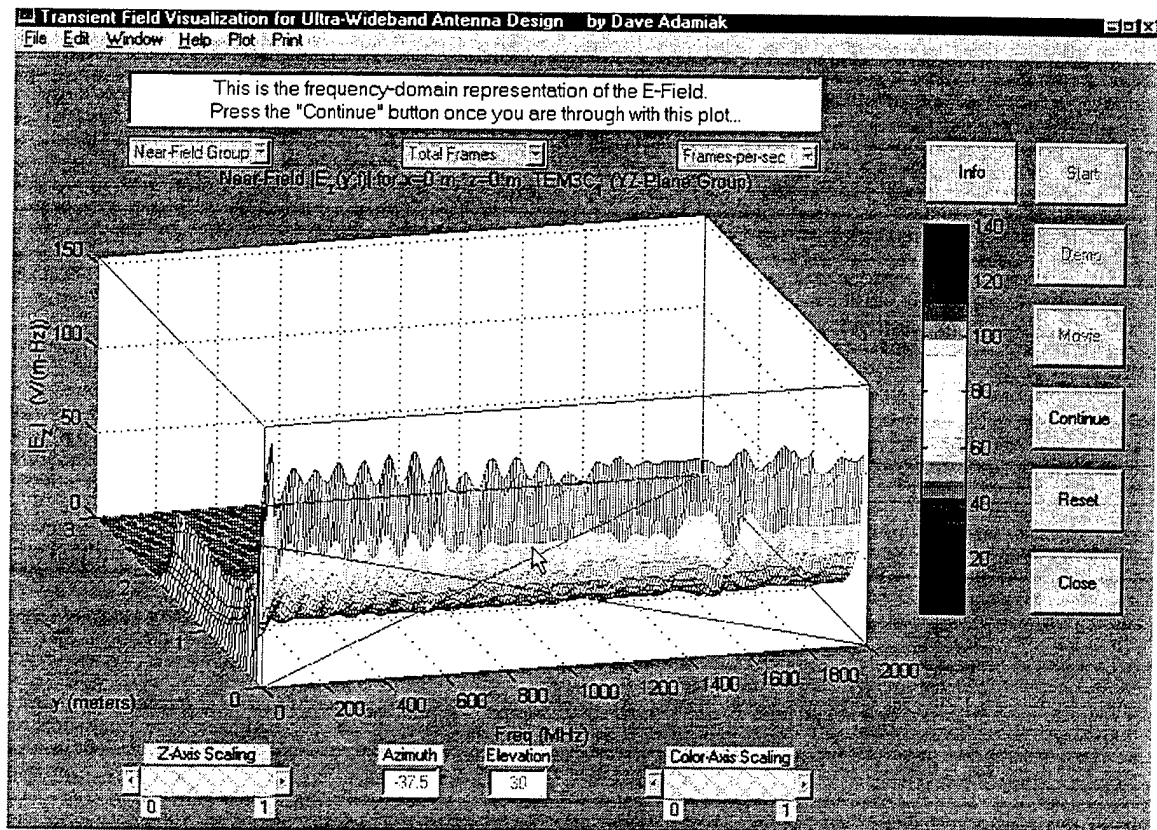


Figure 15. Example of 3-D Plot As View Angle Is Altered

In all, each of the plots allows the user to view the near fields along the centerline axis of the antenna for each frequency both before and after the Gaussian impulse is applied. The user may further view the input waveform directly, too. It should be noted that the fields viewed before application of the impulse are no different from the fields that would occur in the steady-state following application of a 1 Volt sinusoidal voltage signal at the antenna feed.

The frequency domain, while handy in determining the effects the antenna has on the generation of the near fields at each particular frequency, still is not able to answer the questions of "how much of the pulse's energy actually couples and propagates beyond the antenna?", and "to what degree is the pulse's waveform altered as it propagates through and out of the antenna?". These, and other questions, are best determined in the time domain, and that is where the "Movie" subroutine plays an integral role in the analysis process.

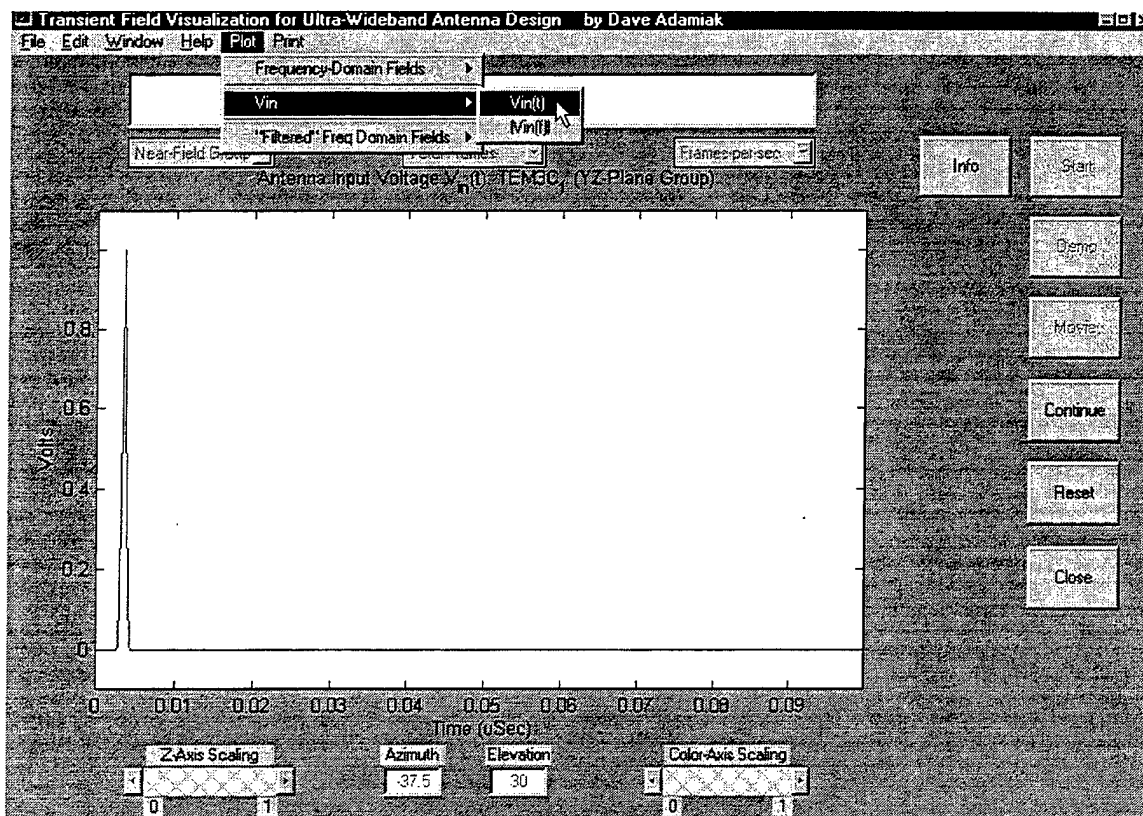


Figure 16. Example of Gaussian Input Voltage Plot

F. MOVIE SUBROUTINE

Once the “Start” subroutine is complete, or at anytime there exists valid data in either the default output *.MAT* files of Section 4.A.4 or other user-named files of the same configuration and design, the user may create a movie to visualize the near fields in the time domain. After depressing the “Movie” push-button, the user is asked to choose between creating a new movie of the time-domain fields or to replay a movie that had previously been created and saved. We will begin the discussion with the methodology for creating a new movie and hold off on the discussion of replaying previously-created movies until the end of this section.

After electing to create a new movie, the user is asked to select the file which contains the desired component of the time-domain fields and the associated header file

that contains the minor variables needed to work with all three of the time-domain fields here. The software prompts the user with the correct default filenames for each *.MAT* file; however, if these files were manually renamed by the user earlier, then those files, too, can be loaded here in place of the default *.MAT* files.

After the files are loaded, the user is shown the values for three variables: T , N , and dt . The variable T is the total time duration of the movie, in μsec . It does not represent the time it will take to view the movie, but rather the timeframe that the movie is depicting. The movie is similar to a slow-motion instant replay of the effects of the Gaussian input pulse over the entire timeframe T . N represents the maximum number of frames the user may use in designing the movie. It also represents the total number of time steps between 0 and T . Finally, the variable dt represents the length of each time step (in nsec). At the same time the values of these variables are shown to the user, the software also prompts the user to first select the desired number of frames for the movie from the “Total Frames” pop-up menu, and then select the frame rate (in frames-per-second, fps) from the “Frames-per-sec” pop-up menu.

After storing that information, the software then renders a sample frame from the movie, as depicted in Figure 17. Its purpose is to provide a “snapshot” from the movie for use in tailoring optimum/desired viewing angle and scale factors. Like the 3-D frequency-domain plots of Figure 14 and 15, the 3-D image projected here can also be rotated freely about itself in the manner described in Section 4.A.5. Unlike there, the azimuth and elevation windows at the bottom of the GUI will display the current view angle after every complete update cycle (after the “Continue” button is pressed). In addition, this figure can be scaled along the z -axis in order to provide better resolution or greater emphasis upon a specific area of the pulse of the antenna. The z -axis scaling “slider” is located at the bottom left corner of the GUI. It has a range of 0 to 1, representing 0 to 100% of the maximum value of the particular near field, since the default z -axis scale is set to the maximum value of the particular near field being plotted. This value will always occur near the feed, since the cross-sectional area of the antenna at that point is the smallest. Therefore, it is

recommended that for use with any thin antennas, the z-axis should be scaled down. Reducing the scale of the z-axis does not change the data that will be plotted; it merely alters the range of the z-axis that will be plotted within the entire plot-box, or display. Figure 18 displays this, and, as compared to Figure 17, it is easy to observe that the Gaussian pulse is much more clearly defined. Also, as with the azimuth and elevation windows at the bottom of the GUI, the rendering of the plot will only be updated with the new z-axis scale **after** the "Continue" button is pressed. For this example, the view angle was also modified to provide a better side aspect of the pulse as it propagates through the antenna.

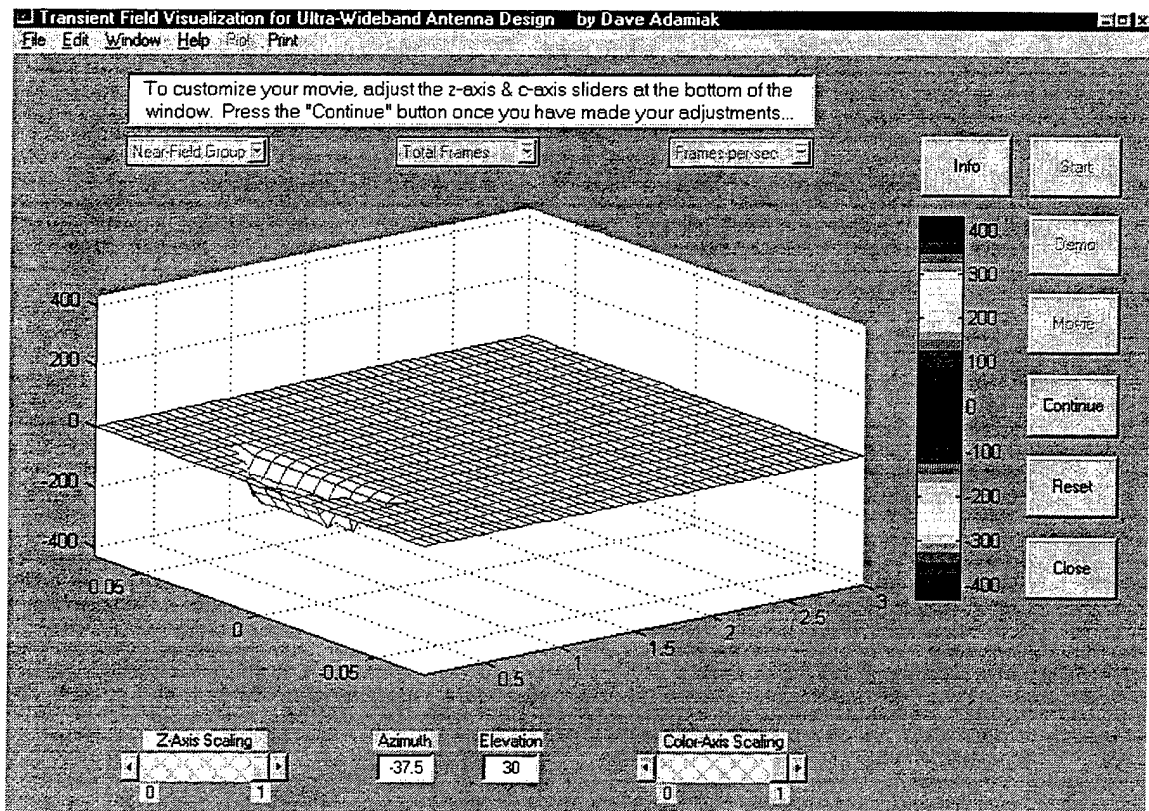


Figure 17. Example of Initial Sample Frame from Movie Subroutine

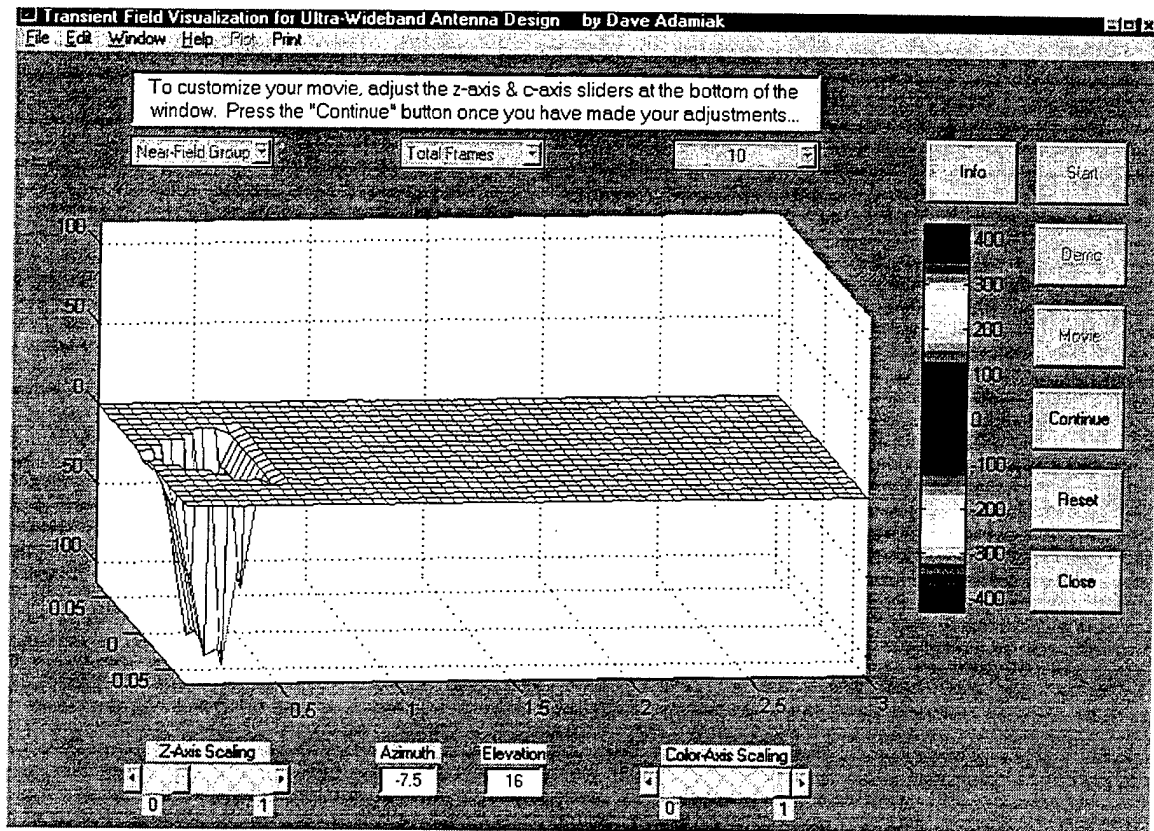


Figure 18. Sample Frame from Movie after Modification of Z-Axis and View Angle

Another variable that can aid in better graphical analysis of the time-domain fields is the modification of the color-axis (c-axis). By adjusting the “Color-Axis Scaling” slider at the bottom of the GUI, the user is able to scale the range of the numerical values assigned to the colormap. The colormap, as depicted in the colorbar to the right of the plot in Figure 18, consists of 128 separate colors which range from “hot” red at the minimum value to “cold” blue at 0 and back to “hot” red at the maximum value. The default values assigned to the colormap are the same as those assigned to the z-axis. In much the same manner as for the z-axis scaling, then, scaling of the c-axis will provide better resolution of a particular area. In the case of color, though, a subtle change in the amplitude of the pulse as it propagates can produce a dramatic color shift, depending upon how well the c-axis is scaled. Figure 19 is our original sample plot with the c-axis scaled for better emphasis of the pulse. By scaling the z-axis and c-axis separately, the user may keep a majority of the

maximum value of the pulse within the figure window while also emphasizing an area of interest with the color palette.

As further seen in Figure 19, after the user updates/changes the sample plot and depresses the “Continue” button, the software will automatically update the plot and then ask the user if he would like to modify the plot again. Until the user selects “No”, the process of modification and updating of the plot will continue ad infinitum. Once the plot is completely customized to the user’s desires and the adjustments cease, the software then builds the movie frame-by-frame, plotting the near-fields for each time step. Once all requested frames are constructed, the movie will run automatically at 12fps and then at the frame rate selected by the user. After the movie runs through the playback mode (at both speeds) once, the user may select whether to replay the movie as many times as he/she wishes.

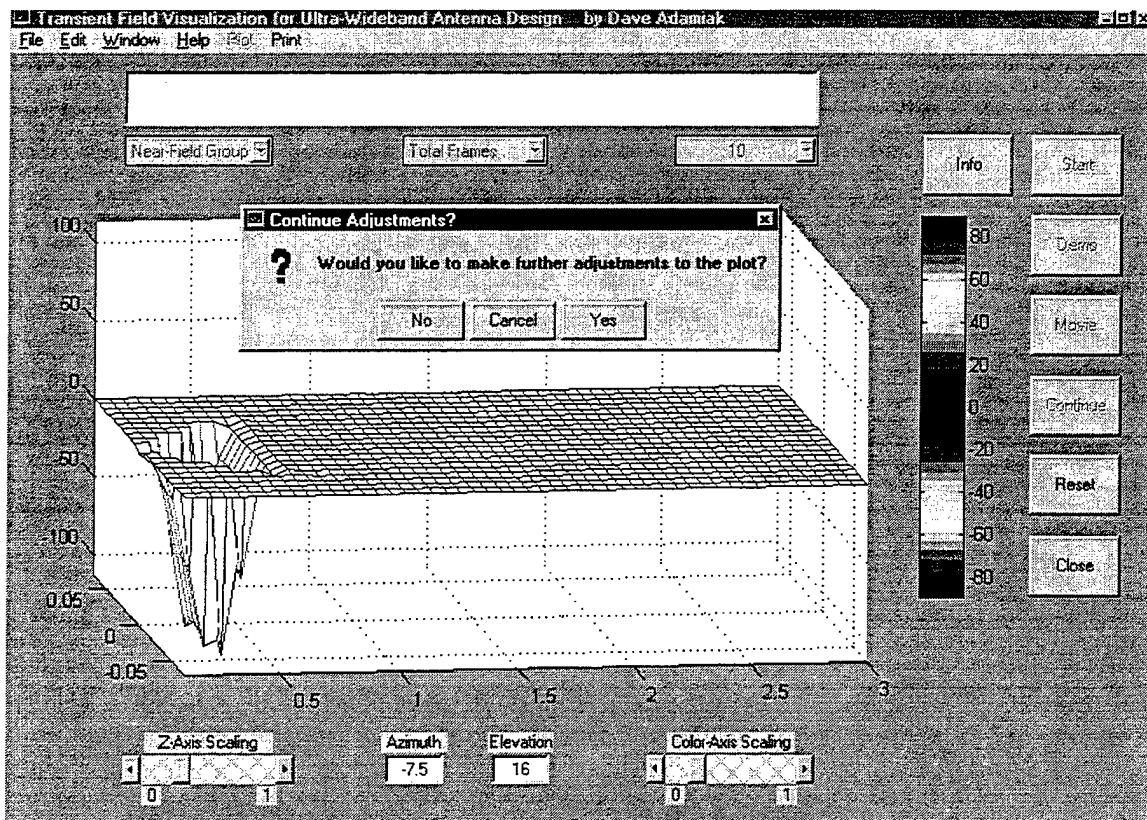


Figure 19. Sample Frame from Movie after Modification of the C-Axis

After the final replay, the user once again is given the option to make adjustments to any of the movie parameters and rebuild it again. If this option is selected, the movie is erased from memory, and the sample frame with current view angle and default scaling factors is again displayed for modification as outlined earlier. If the rebuild option is not selected, the user is then prompted to provide a filename under which to save the movie for replay at a later time.

In order to replay a movie which has been previously created and saved, the user need only to depress the "Movie" push-button anytime it is not deactivated. The user is then prompted to select the filename under which the movie may be found and further asked to select a frame rate. Once that is done, the software loads the movie and runs it twice - once at 12fps, and once at the frame rate selected by the user. The user can replay the movie ad infinitum without reloading it each time, but no parameters may be changed, since the saved movie is merely a raster-scan image (like a photograph) of each frame. This prevents the movie axes or view angle from being modified.

The movie subroutine is, by far, the more robust tool of this software package. Its importance should become even more evident as we move to analyze specific antenna structures on the way to designing an UWB antenna that will successfully reproduce a received impulse waveform with near-perfect fidelity in the frequency range of 10 MHz to 10 GHz.

V. VALIDATION OF THE SOFTWARE AND ANALYSIS OF THE PROTOTYPE TEM HORN

In order to first demonstrate the accuracy of the software and to secondly validate an UWB antenna design meeting certain parameters, various antenna geometries were developed and fully tested using the *AWAS* and *NFLDVIS.M* analysis package. A summary of each geometry, its purpose in the validation stage, and its performance as a wideband or UWB antenna is provided in the next section. Following that section, the prototype TEM Horn is then analyzed as an UWB antenna candidate.

A. SIMPLE WIRE ANTENNA STRUCTURES

To validate the accuracy of the software package, several basic two-wire antennas were created and tested. The specific behavior of each antenna structure presented in this section could be determined from known radiation characteristics of the simple two-wire dipole and the “vee” dipole, and from transmission line theory. The fundamental properties of dipole antennas are both well-known and easily demonstrated using basic antenna theory. As a result, they lend themselves well towards validation/verification of the software.

As a point of reference for comparison between these antennas, the near-field values for each of the simple wire structures were calculated over a horizontal plane “cut” through the centerline of each antenna. The physical dimensions of the plane extended to two meters on either side of the antenna (along the x -axis) in one dimension, and from the antenna feed to a point 1-to-2 meters beyond the antenna aperture in the other dimension. Because of this, the perspective presented to the viewer of the time-domain movie is as if he/she is standing at the antenna feed and looking straight down the centerline axis of the antenna (the positive y -axis).

1. The Simple Dipole

The dipole, as shown in Figure 20, consists of two equal length arms extending from the origin along the positive and negative z -axis. Dipoles are resonant structures, since the current wave travelling outward from the feed point to the end of the wire is reflected, supporting the formation of a standing wave. From basic antenna theory, it is well known that an ideal dipole of length $L \leq \lambda$ oriented along the z -axis will create a radiation pattern that resembles a donut shape surrounding the midpoint of the dipole, as illustrated in Figure 21 [2]. This pattern can be considered omnidirectional, since it is uniform in the xy -plane (the H-plane of this antenna). Further, it can be shown that whenever the length, L , of the dipole is an odd multiple of half the operating wavelength, i.e.

$$L = \frac{n\lambda}{2} \quad (n = 1, 3, 5, \dots), \quad (5.1)$$

the antenna will always radiate some portion of its power in the H-plane (the xy -plane for the antenna depicted in Figure 20). The symmetry observed in the radiation pattern of the dipole should then infer an expectation of symmetry in the time-domain E-field plot

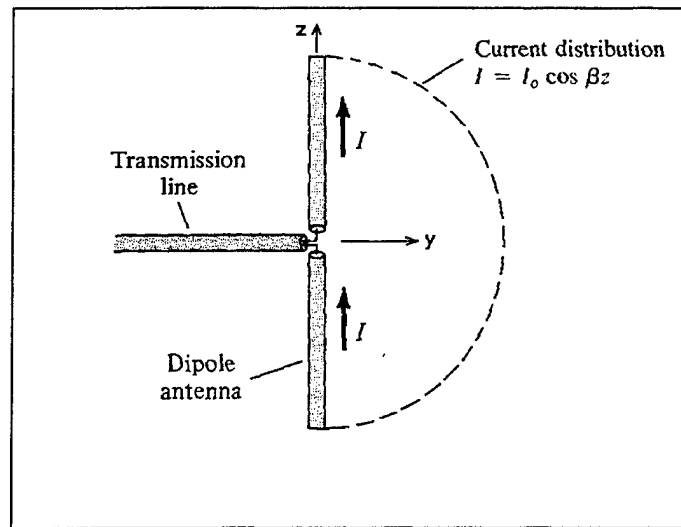


Figure 20. A Half-Wave Dipole and Its Current Distribution

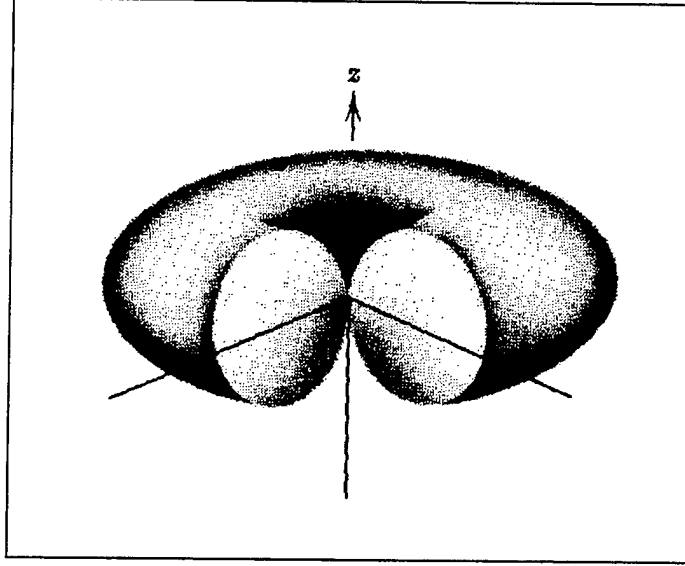


Figure 21. Radiation Pattern for an Ideal Dipole

displayed by the software, and thus, the application of an impulse voltage to the dipole should result in a time-domain E-field which expands radially in the H-plane.

For the dipole geometry shown in Figure 22, the theoretical z-component of the E-field can be expressed as

$$E_z = -j30I_m \left[\frac{e^{-jkR_1}}{R_1} + \frac{e^{-jkR_2}}{R_2} - 2 \cos(kl_1) \frac{e^{-jkr}}{r} \right], \quad (5.2)$$

and if we limit ourselves to measuring the field values along the y-axis, only, then Equation (5.2) simplifies to

$$E_z = -j60I_m \left[\frac{e^{-jkR}}{R} - \cos(kl_1) \frac{e^{-jkr}}{r} \right], \quad (5.3)$$

where I_m is the maximum amplitude of the sinusoidal current distribution on the dipole; $k = \omega / c = 2\pi / \lambda$ is the propagation constant; $R = R_1 = R_2 = \sqrt{r^2 + l_1^2}$ is the distance from the endpoints of the dipole to the arbitrary observation point $P(0, y, 0)$ along the y-axis; $l_1 = L/2$ is the length of each arm of the dipole; and $r = \rho$ is the radial distance from the dipole feed to the field observation point P [8]. Of note is the fact that the last

term of Equations (5.2) and (5.3) vanishes whenever $L = n\lambda / 2$ for $n = 1, 3, 5, \dots$ because $\cos(kl_1) = \cos(n\pi / 2) = 0$, for $n = 1, 3, 5, \dots$

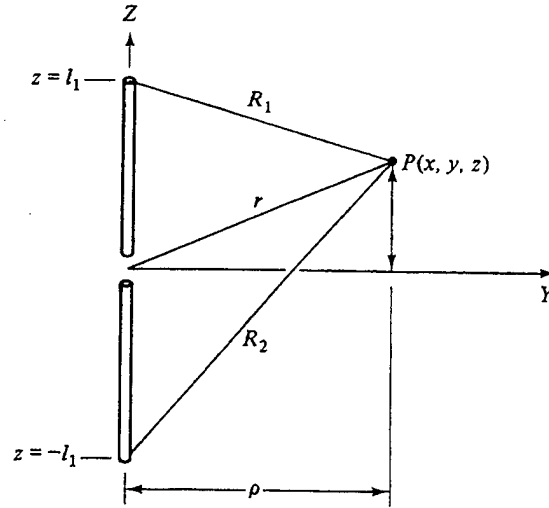


Figure 22. Geometry Used in Dipole E-Field Calculations. From [8].

Created using the MATLAB code found in Appendix B, Figure 23 is a plot of the magnitude of E_z as a function of both frequency and position along the y-axis. Most obvious is the periodic pattern of maximum and minimum field values that occurs over the entire frequency range at a distance of approximately 0.1 meters from the dipole feed point. If we solve for the wavelength, λ , where the half-wave dipole condition is met, i.e. where $L = \lambda/2$, we find that for $L=2.06156$ meters (the total length of our dipole), a peak field value should occur at a wavelength of 4.12312 meters, or a frequency of 72.76 MHz. Not coincidentally, then, this is approximately the frequency where we observe our largest peak in the E-field plot of Figure 23. In addition, we are able to determine upon closer inspection of the plots that all of the successive peaks occur precisely 145.52 MHz apart. This frequency corresponds to the inverse of the roundtrip time it takes a pulse to travel from the feed of the dipole, to one of the endpoints, and back. This repetition frequency is equal to the resonant frequency of the dipole [2].

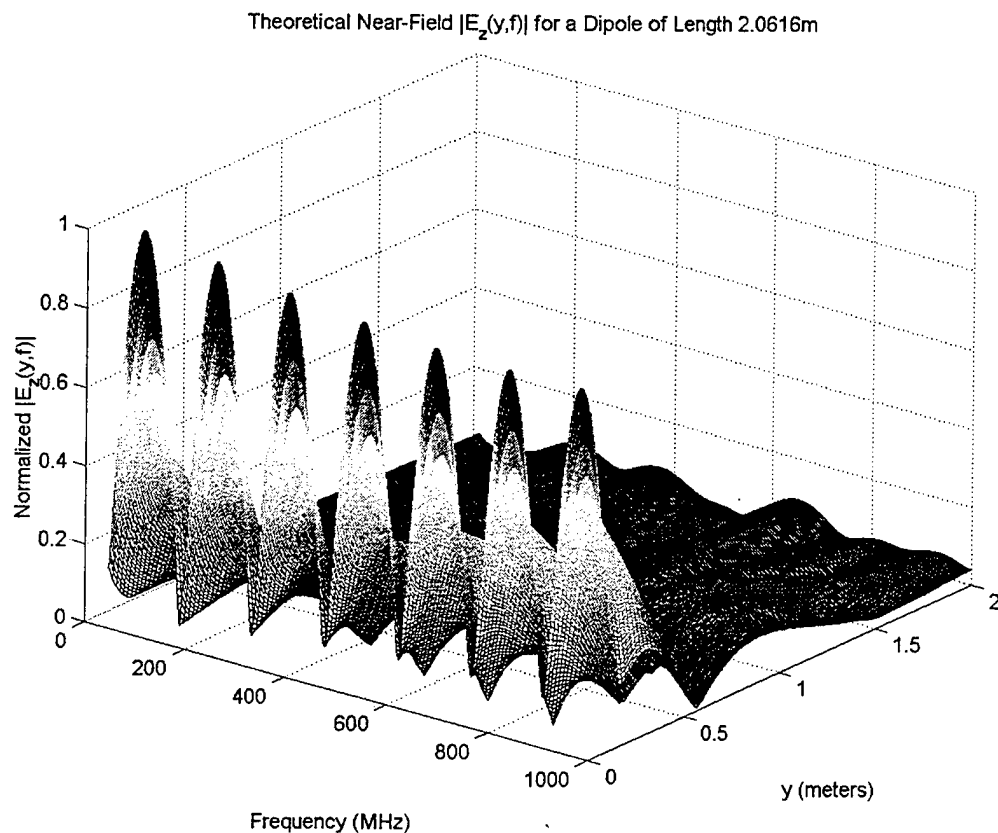


Figure 23. The Theoretical Near Electric Field, $|E_z(y,f)|$, of a Simple Dipole

Returning to Figure 23, for a given fixed frequency, we note that an interference pattern develops as the distance from the feedpoint is increased. Figure 24 more clearly illustrates this interference pattern. Considering the fact that the simple dipole can be modeled as the superposition of three point sources (one at each end of the dipole and one at the feed point), it is clear that this pattern develops as a result of the constructive and destructive interference of the fields produced by the three sources. As an initial test of the visualization software, it then makes sense to construct a dipole of similar dimensions and compare the results to theory.

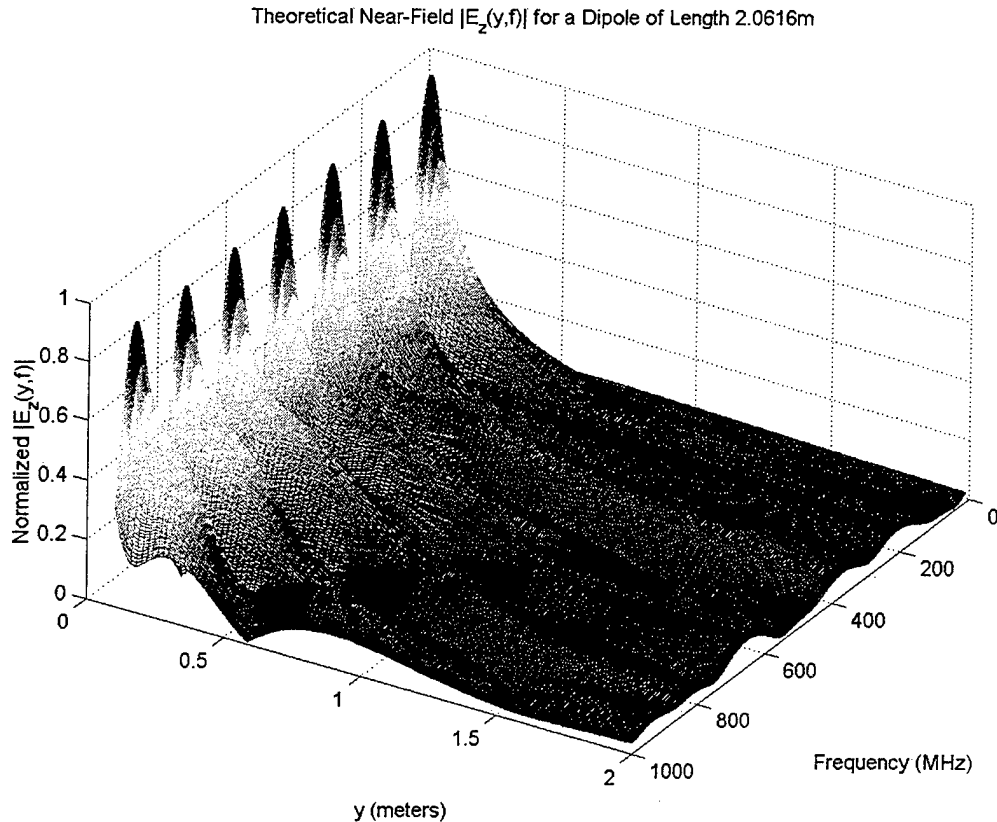


Figure 24. The Theoretical Interference Pattern in the Near Field of the Simple Dipole

After creating the dipole in *AWAS* and calculating the frequency- and time-domain near-fields, Figure 25 was created by the software for the “steady-state” condition, i.e. the representation of the fields after a unity amplitude sinusoidal voltage signal was applied to the antenna feed. Upon quick inspection, one should notice several “peaks” in the E-field plot. Not coincidentally, a distinct peak E-field value is at approximately 70 MHz, and several others appear approximately every 150 MHz after that. Theoretically, we expected the first peak to occur at 72.76 MHz, with each successive one precisely 145.52 MHz apart. So far, the fields generated by the software are consistent with theory. One primary difference, though, between the fields calculated by the Method of Moments and those derived from Equation (5.3) lies in the lack of distinct minima in the region 0.1 meters from the feed. This is due to *AWAS*’s use of fewer discrete frequencies in solving for the fields and also the fact that the theoretical

calculations ignore such real-life phenomena as fringe effects and assume ideal current distributions along the entire length of the dipole.

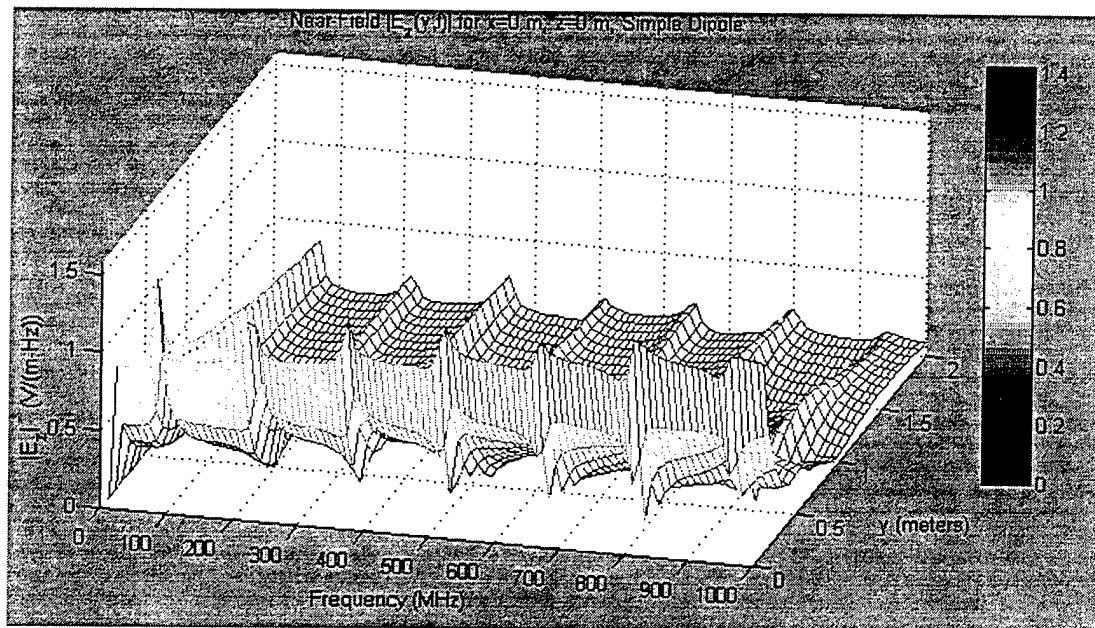


Figure 25. Plot of E-Field Magnitude vs. Frequency and Position Along the Y-Axis for the Simple Dipole

Also, when we look at this plot from the perspective shown in Figure 26, we are able to closely examine the interference pattern within the E-field for comparison with the theory illustrated by Figure 24. Over the entire frequency range, the interference pattern illustrated in Figure 26 is consistent with the theoretical pattern discussed earlier. All that remains, then, is a closer examination of the results generated by both AWAS and the theoretical model at specific frequencies.

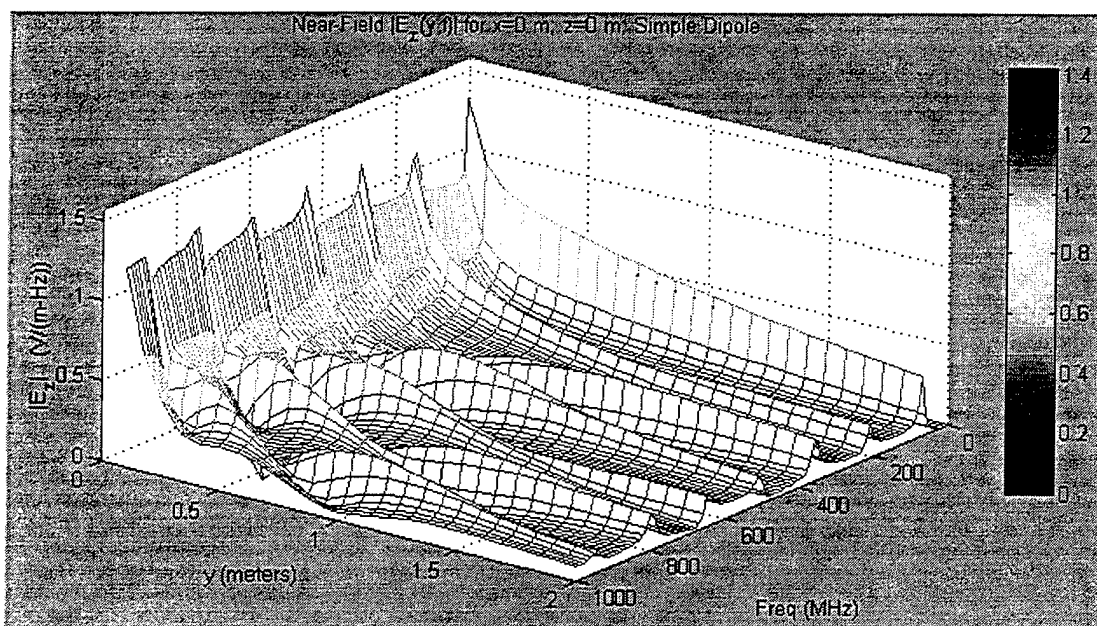


Figure 26. Plot of E-Field Magnitude as a Function of Distance vs. Frequency

Using the MATLAB code of Appendix B, a direct comparison of AWAS versus theory was made at frequencies of 36.38 MHz, 72.76 MHz, and 218.281 MHz. For the given dipole length of 2.06156m, these correspond to the frequencies where the dipole is electrically a quarter-wavelength, a half-wavelength, and three-halves of a wavelength long. The following figures are comparisons of the magnitude of E_z , its real component, $\text{Re}\{E_z\}$, and its imaginary component, $\text{Im}\{E_z\}$, as functions of position along the y-axis

In Figure 27, the AWAS-derived solution for the E-field is almost exact in comparison with the theoretical values. This indicates that the theoretical approximation of the near-fields is obviously a good one for an electrically-short dipole. It is also indicative of the accuracy of AWAS in calculating the near E-fields. Additionally, it is evident that the two solutions converge as the observation point is moved further from the antenna. Figures 28 and 29 are the comparisons of the real and imaginary components of the field, respectively. As expected, they, too, indicate strong correlation between AWAS and theory.

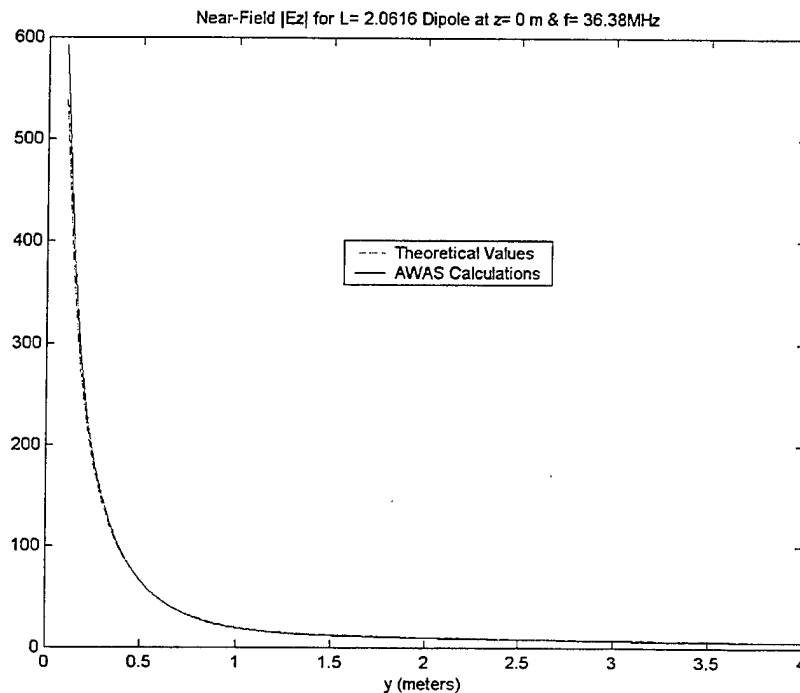


Figure 27. Comparison Between AWAS vs. Theoretical $|E_z|$ for a $\lambda/4$ Dipole

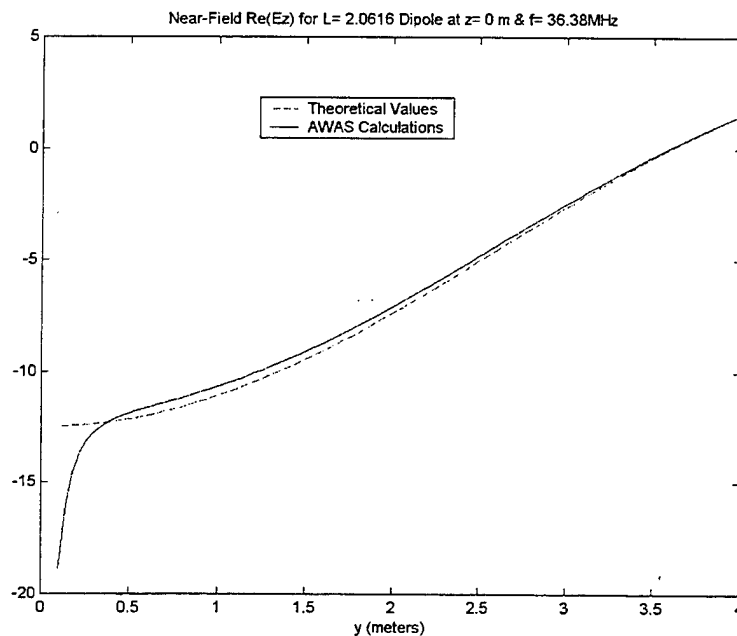


Figure 28. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $\lambda/4$ Dipole

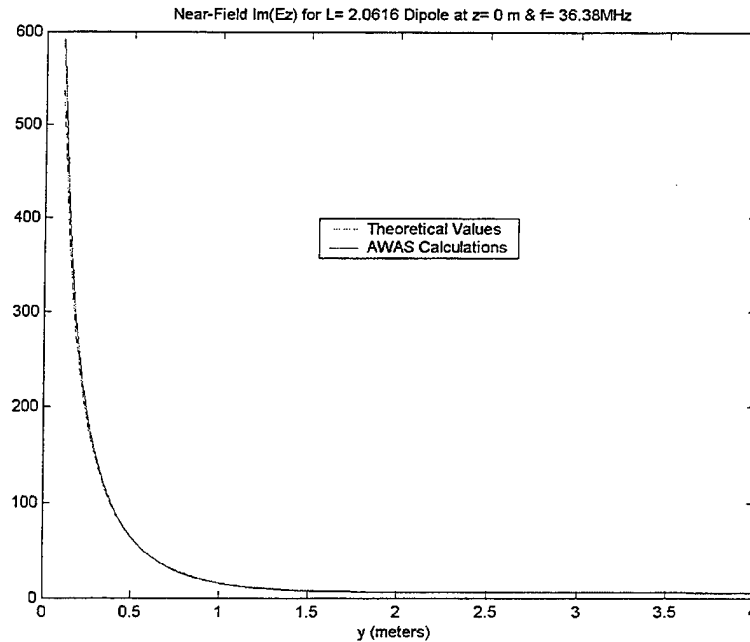


Figure 29. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $\lambda/4$ Dipole

For the half-wave dipole case, Figures 30 through 32 again clearly show good correlation between the AWAS model and theory. As the electrical length of the dipole increases, though, the margin of error between the two also increases. As noted previously, it is the theoretical solution which is losing accuracy, and not the Moment Method solution. The margin of error is even more significant in the $3\lambda/2$ dipole case, as shown in Figures 33 through 35; however, most of the error is merely amplitude error, with the general “shape” of the theoretical and AWAS-calculated fields remaining well-correlated.

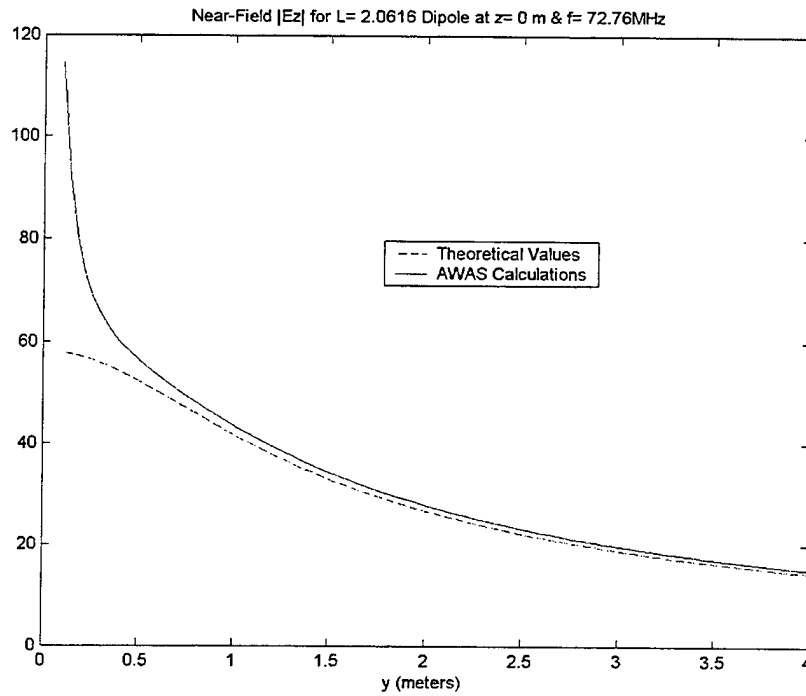


Figure 30. Comparison Between AWAS vs. Theoretical $|E_z|$ for a $\lambda/2$ Dipole

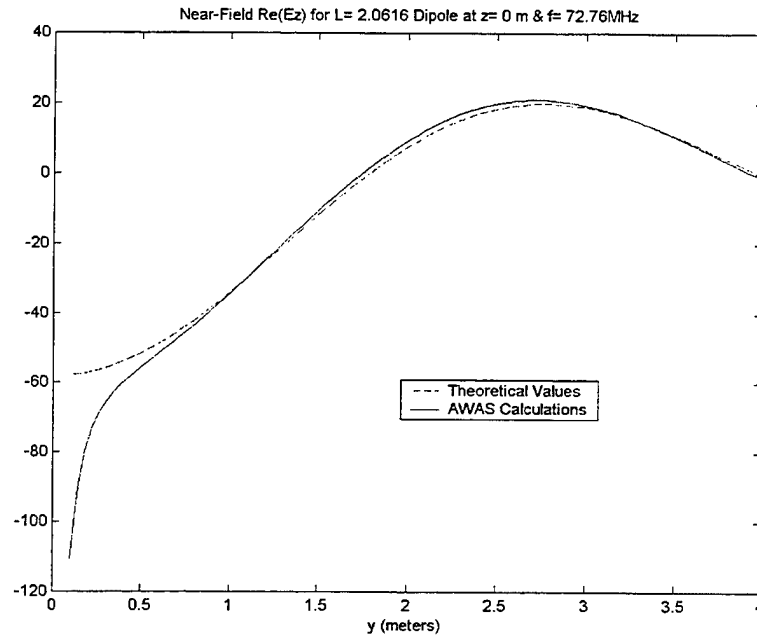


Figure 31. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $\lambda/2$ Dipole

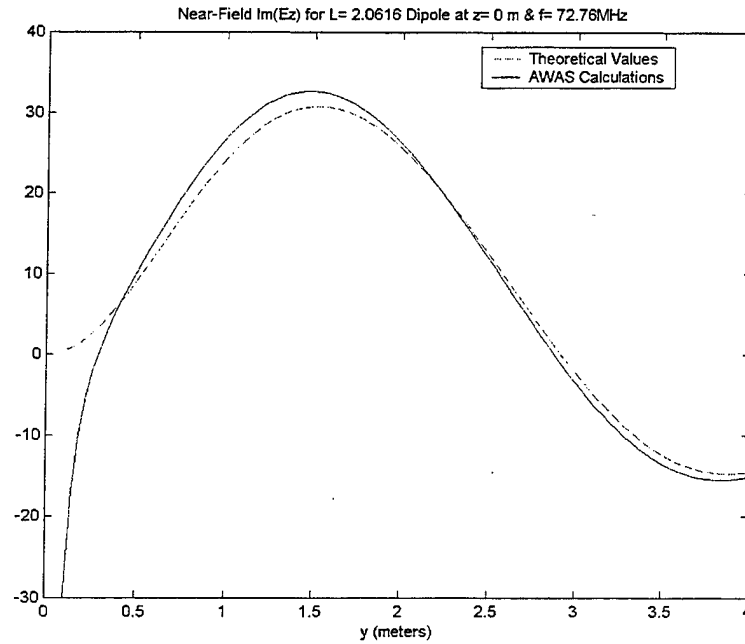


Figure 32. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $\lambda/2$ Dipole

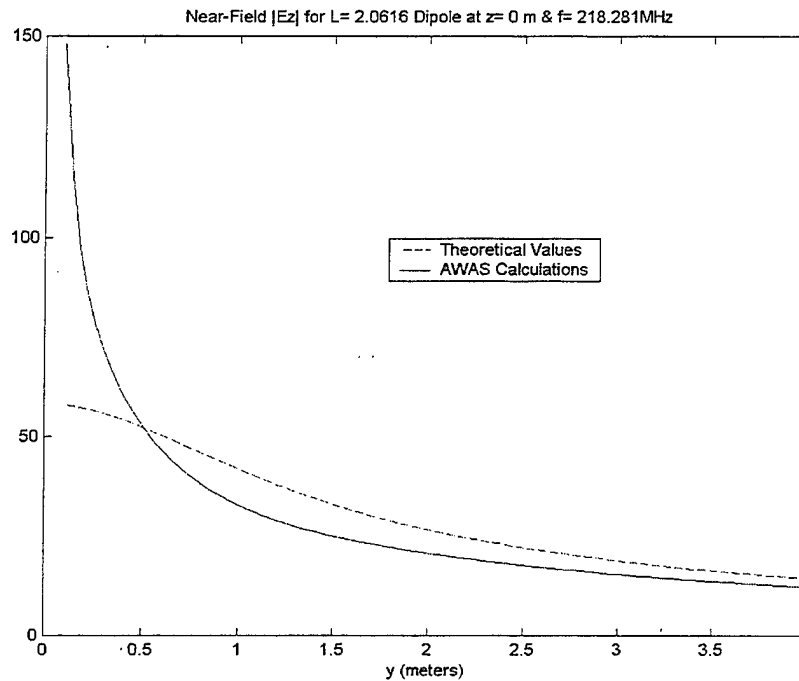


Figure 33. Comparison Between AWAS vs. Theoretical $|E_z|$ for a $3\lambda/2$ Dipole

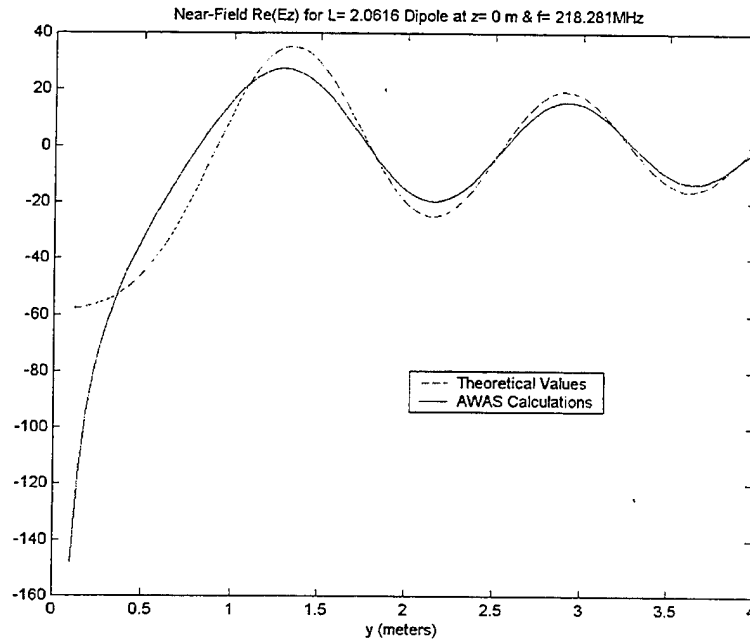


Figure 34. Comparison Between AWAS vs. Theoretical $\text{Re}\{E_z\}$ for a $3\lambda/2$ Dipole

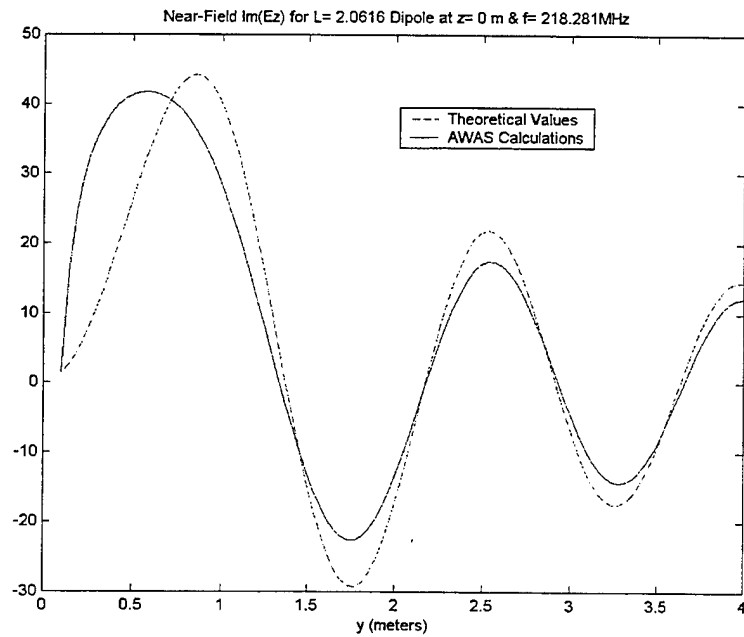


Figure 35. Comparison Between AWAS vs. Theoretical $\text{Im}\{E_z\}$ for a $3\lambda/2$ Dipole

With the above observations validated by our theoretical model, we now turn our attention to the time-domain representation of the dipole's near E-field. Specifically, since we are viewing the propagation in the H-plane, we look for a radially symmetrical propagation pattern due to our Gaussian impulse. In lay terms, what we expect to see is an image similar to that seen by dropping a rock into a still pond: an ever-expanding circular propagation of a wave. As seen in Figure 36, this is most certainly the case. One thing to note about the figure is that the additional wave being generated behind the first one is not due to an additional input impulse, but rather, due to current reflections along the length of the antenna which partially launch into free space on each round trip along the structure. Note, too, that the period between "waves" is approximately 2 meters. A repetition frequency of 145.521 MHz means a period between reflections of 6.872 nsec. At the speed of light, an RF pulse travels 2.0616 meters in 6.872 nsec, which coincides exactly with the distance between the original Gaussian impulse and the first reflection in Figure 36.

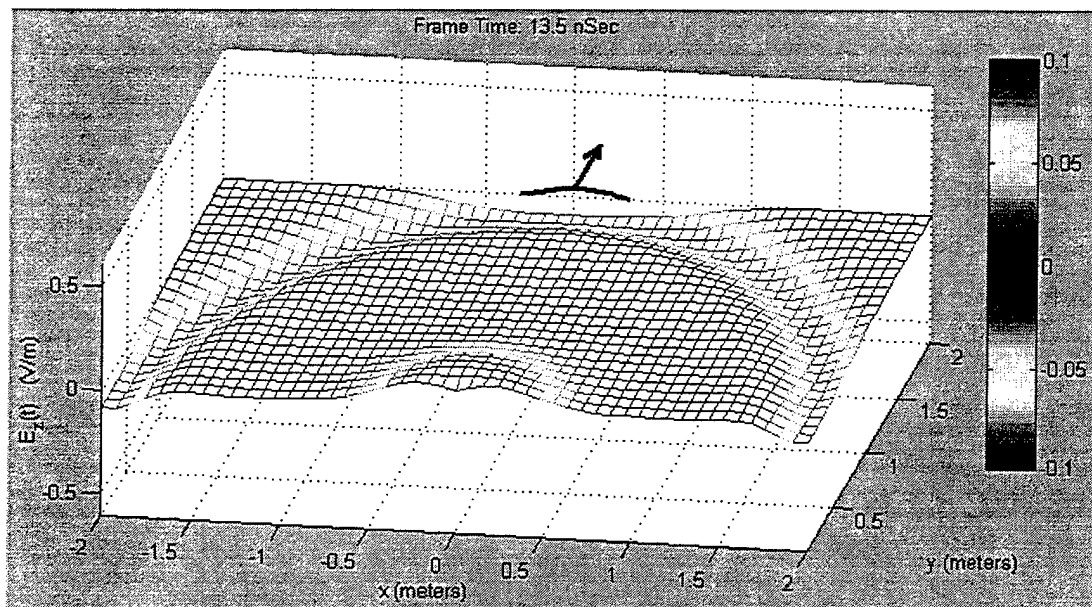


Figure 36. A Snapshot of the Time-Domain E-Field of a Simple Dipole Transmitter at $t=13.5\text{nsec}$

To graphically illustrate the Lorentz Reciprocity Theorem discussed in Chapter II, we can look again at the time-domain E-field of the simple dipole as a receiving antenna. For this case, we use a double-Gaussian impulse as the incident wave, propagating in the $-y$ direction (as shown in Figure 37). This incident field pulse shape has an amplitude of 1 V/m and contains no dc (a single Gaussian pulse does). The incident wave impacts the dipole and induces a current along the antenna structure. As the current proceeds along the dipole, it produces a reflected field (Figure 38) which is the same as in the transmitting dipole case (Figure 36), with the exception of a scale factor. In addition, the resonance of the dipole over such a large frequency range guarantees that we get the same current reflections, which partially launch into free space on each roundtrip along the structure. Thus, either the transmitting antenna or receiving antenna can be used interchangeably in any discussion of the antenna's near fields.

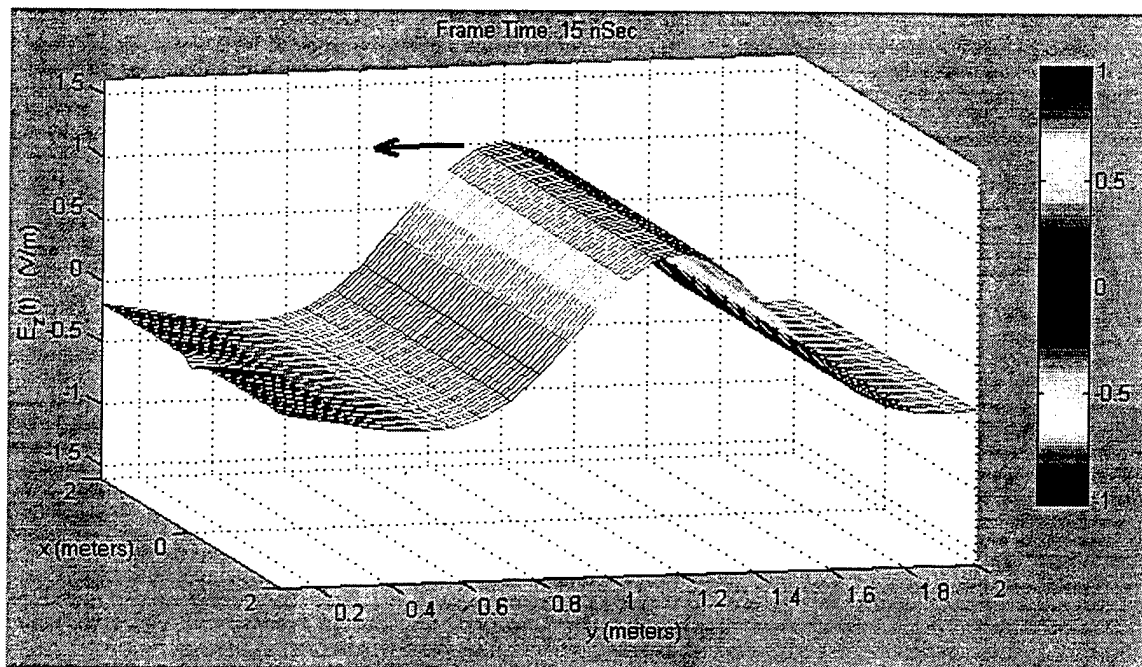


Figure 37. Incident E-Field Generated by a Double-Gaussian Impulse at $t=15\text{nsec}$

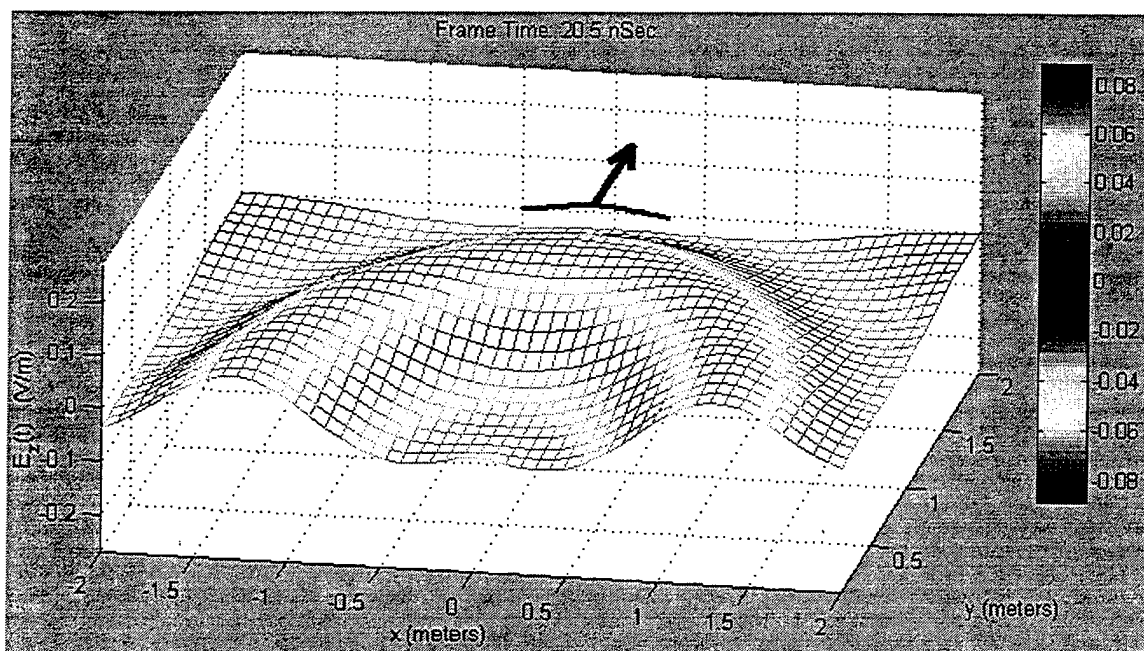


Figure 38. The Time-Domain E-Field of a Simple Dipole Receiver at $t=20.5\text{nsec}$

Regardless of whether the simple dipole is analyzed as a transmitting antenna or directly as a receiving antenna, it is clear that both methods of analysis accurately determine that the simple dipole is a poor UWB receiving antenna. In the frequency-domain depictions of the dipole's near E-field (Figures 25 and 26), the magnitude plot of the near E-field displays resonant peaks at a number of frequencies. These resonant modes will cause amplitude distortion of the input impulse signal. In addition, the time-domain depictions (Figures 36 and 37) clearly indicate the presence of current reflections throughout the dipole. These reflections are indicative of a nonlinear phase change, since it is the lower frequency components of the impulse that we view "ringing" back and forth along the dipole. Thus, the simple dipole causes phase distortion of the input impulse, too, which makes it an unsatisfactory choice as an effective UWB antenna.

Using the simple dipole model, it has been shown that the software does, in fact, accurately display the electromagnetic phenomena associated with the application of input pulses to a dipole antenna. To ensure the software is robust and able to handle

complex antennas, we now move on to test it with some “special case” dipole antennas and modified two-wire transmission lines.

2. The Vee Dipole

The Vee Dipole is best thought of as a simple dipole whose ends are bent towards the y -axis to form an angle, γ , as shown in Figure 39. It can also be thought of as an open-ended transmission line whose ends are bent toward the z -axis. Because of its unique shape, the directivity of such an antenna will be greater than that for a straight dipole, since the radiation pattern is merely the superposition of that from each straight-wire section [2]. Thus, the Vee Dipole is able to focus substantially more power down its boresight axis, and thereby reduce the sidelobe field levels. For the Vee Dipole shown, the corresponding directivity is

$$D = 2.94(h/\lambda) + 1.15, \quad (5.4)$$

where h is the length of each arm of the dipole ($h = L/2 = 1.03078\text{m}$ for our example)

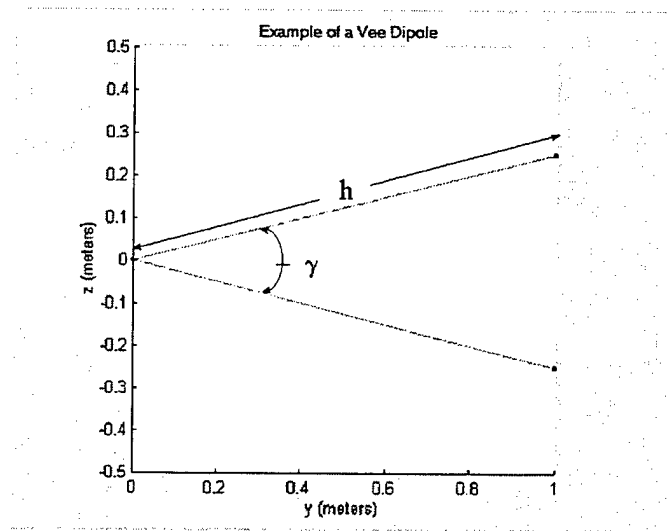


Figure 39. Example of a Vee Dipole

[2]. This equation is valid only over the frequency range 145.5 MHz to 873 MHz for our specific antenna. Regardless, it is expected that over the entire tested frequency range, more energy will be focused along the centerline axis than elsewhere.

To verify the above, a plot of the frequency-domain representation of the E-field along the centerline axis (y -axis) of the Vee Dipole was rendered. It is replicated here as Figure 40, and as a point of reference within the plot, the antenna aperture is at the $y=1\text{m}$ point on the plot. What should be immediately noticed is that the maximum value of the field strength is found at the frequency meeting the half-wave dipole condition, just as it was for the simple dipole. The field strength in this case, however, is significantly larger. One reason is that within the 0-to-1m distance along the y -axis, the field is somewhat constrained by the physical structure of the antenna, and as such, the field is not able to expand as rapidly as in free space. The other is, of course, the directivity the antenna provides as discussed earlier. In addition, the resonant modes of this antenna are identical to those of the simple dipole tested previously. This is because the antenna arms of the tested Vee Dipole are exactly the same length as those of the simple dipole.

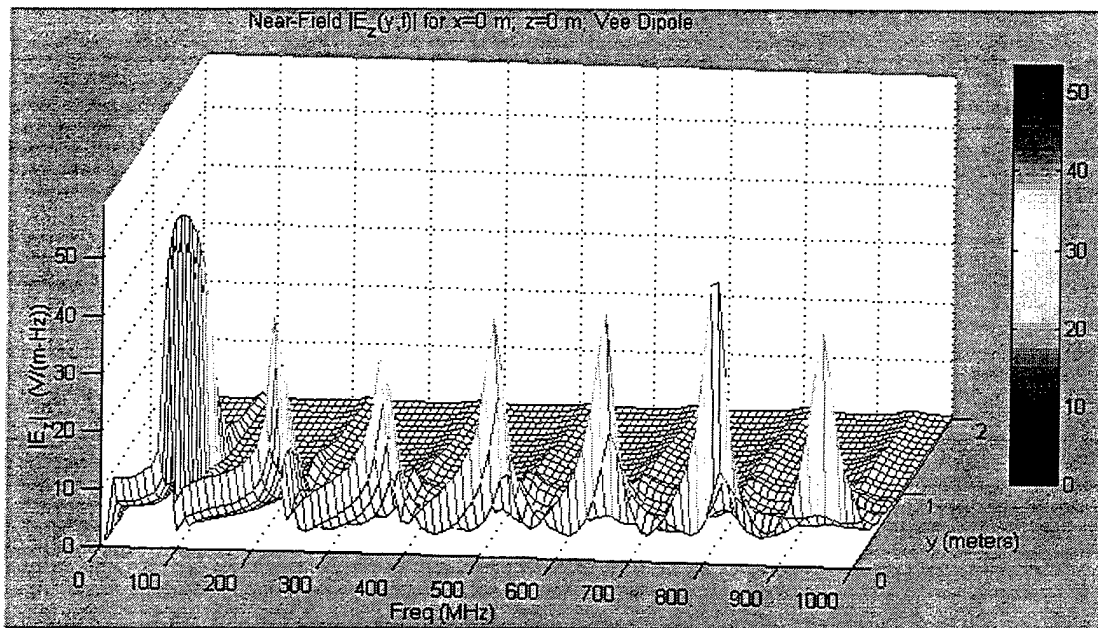


Figure 40. Plot of E-Field Magnitude vs. Frequency and Distance along the Centerline Axis for the Vee Dipole

In Figure 41, we are able to better see the changes to the E-field for each frequency as a function of distance along the centerline axis. In this plot, it is clearly

evident that the field strength at 72.76 MHz falls off less quickly than for any other frequency along the entire length of the antenna. Also, the field strength at this frequency outside of the aperture (in free space) is, indeed, significantly larger than the simple dipole case. Of further note is that for the other frequencies, the field strength within the bounds of the antenna structure takes on an interference pattern with distance that is obviously a function of wavelength. Specifically, we know from transmission line theory that this is similar to the voltage standing wave of the transmission line and that the conditions (a maximum or minimum) repeat themselves every half wavelength. As an example, for a wavelength of 0.317m corresponding to the 946 MHz frequency, we note a distinct maximum at 0.26m along the y-axis. The next maximum then occurs at 0.42m, or approximately 0.16m (a half wavelength) after the first.

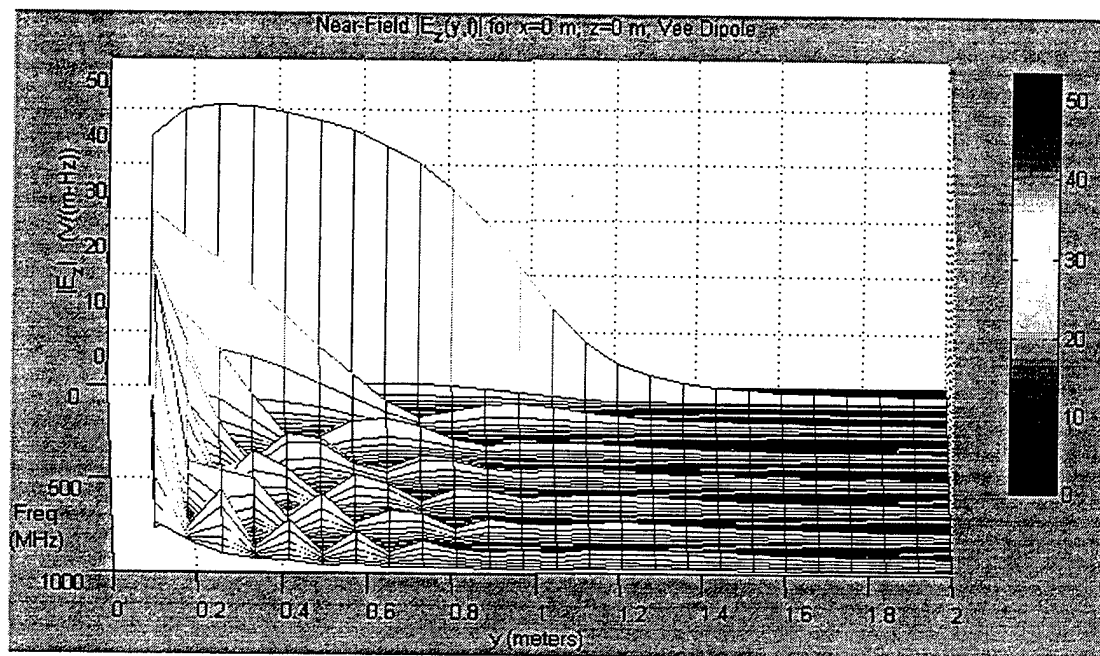


Figure 41. The E-Field along the Vee Dipole's Centerline Axis

In addition, for those frequencies where at least one half-cycle (a maximum and a minimum) occurs within the antenna structure (between 0-to-1m along the y-axis), a large E-field component also denotes resonance (a reflection of the pulse at the aperture). The larger the value of the E-field, as is the case at 218MHz, the more energy is reflected back

from the aperture. In contrast, the lower the value of the E-field, as is the case at 946MHz, the greater coupling efficiency should be observed in the time-domain.

In order to determine whether the antenna's directivity is specifically as theorized above, a time-domain plot is required, since that is where the field strength in more than one direction can be viewed. Figure 42 is a rendering of the time-domain E-field taken 13.5nsec after application of the Gaussian impulse wave. Clearly, the H-plane propagation pattern is of semicircular shape in free space with the greatest field strength displayed along the $+y$ direction. Thus, the directivity enforced by this antenna is confirmed. Additionally, one should note the large disturbance near the origin and along the 0-to-1m point on the y -axis. This is caused by reflections generated by waves unable to immediately couple into free space at the very low frequencies. As they ring back and forth along the antenna, though, they eventually do couple and launch what appear to be additional impulse waves.

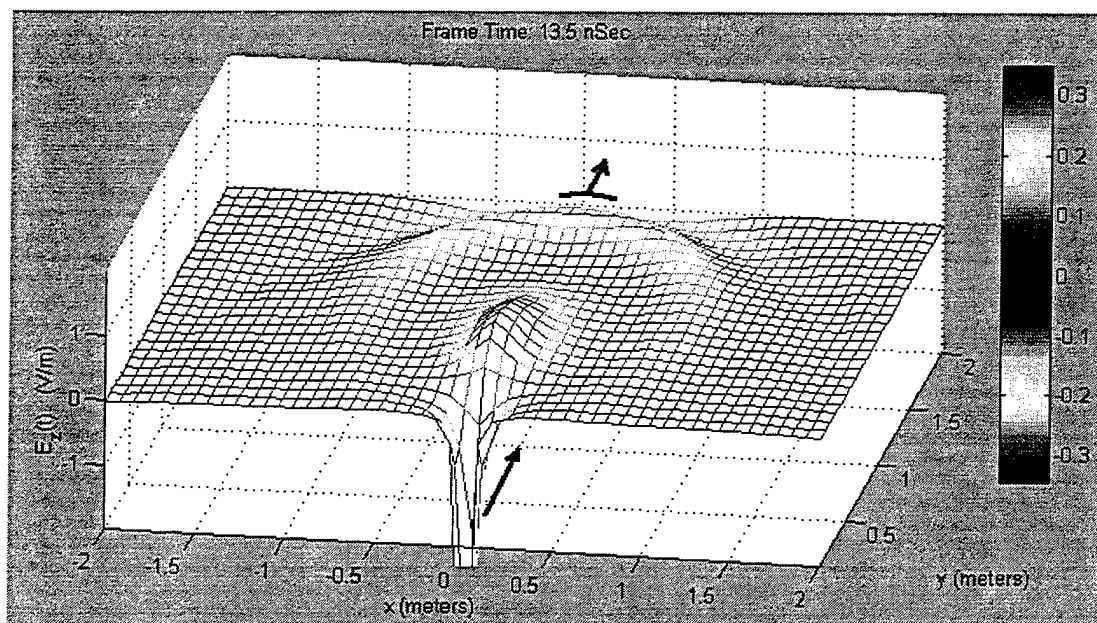


Figure 42. A Snapshot of the Time-Domain E-Field of the Vee Dipole

Like the simple dipole tested earlier, the vee Dipole suffers from the same amplitude and phase distortion problems. Thus, it, too is an unsatisfactory candidate for the role of an UWB receiving antenna.

3. The Vee Dipole with a Parallel Extension

After analyzing the Vee Dipole, a parallel two-wire transmission line was added to the antenna at the aperture, as shown in Figure 43. This structure was chosen next to evaluate the accuracy of the reflection coefficients determined by the software at each end of the transmission line model. Also, the change in the directivity and frequency response of the Vee Dipole due to the extension was also investigated. The first difference that should be observed between this antenna and the first two dipoles is the shift in the wavelength meeting the $L=\lambda/2$ condition due to the increased length of the extension. Using the same equations as in Section 5.A.1, the frequency that meets the half-wave dipole condition for this antenna is approximately 37 MHz.

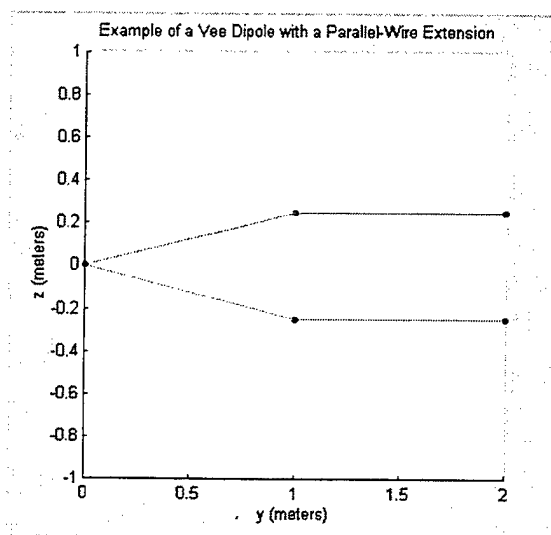


Figure 43. Example of a Vee Dipole with Parallel Extension

It can be shown that the current distribution on a two-wire transmission line will look very similar to the ideal dipole's at certain wavelengths [2]. Because of this, the

time-domain view of the radiation pattern and directivity of this antenna should not differ greatly from the Vee Dipole's. Since we are again viewing the magnitude of the E-field in the H-plane, the wave front should appear less circular than for the Vee Dipole case, and begin to approach a flat planar front because of the extension. The reflection coefficients at the feed and aperture ends of the antenna are similar to those of the Vee Dipole, but because of the additional length, they should be much more discernible in the time-domain plot. Reference [9] predicts a reflection coefficient, Γ , of -1 at the feed and $\Gamma = +1$ at the aperture, which, when multiplied by the value of the incident wave, will provide the value of the reflected wave.

Looking first at the frequency-domain representation of the E-field in Figure 44, it is clear that the E-field at 37 MHz displays better strength than any other frequency through the entire length of the antenna, as was predicted. The resonant frequency of 74 MHz is also clearly evident in the periodicity of the E-field maxima at a distance of $y=0.1\text{m}$ from the feed.

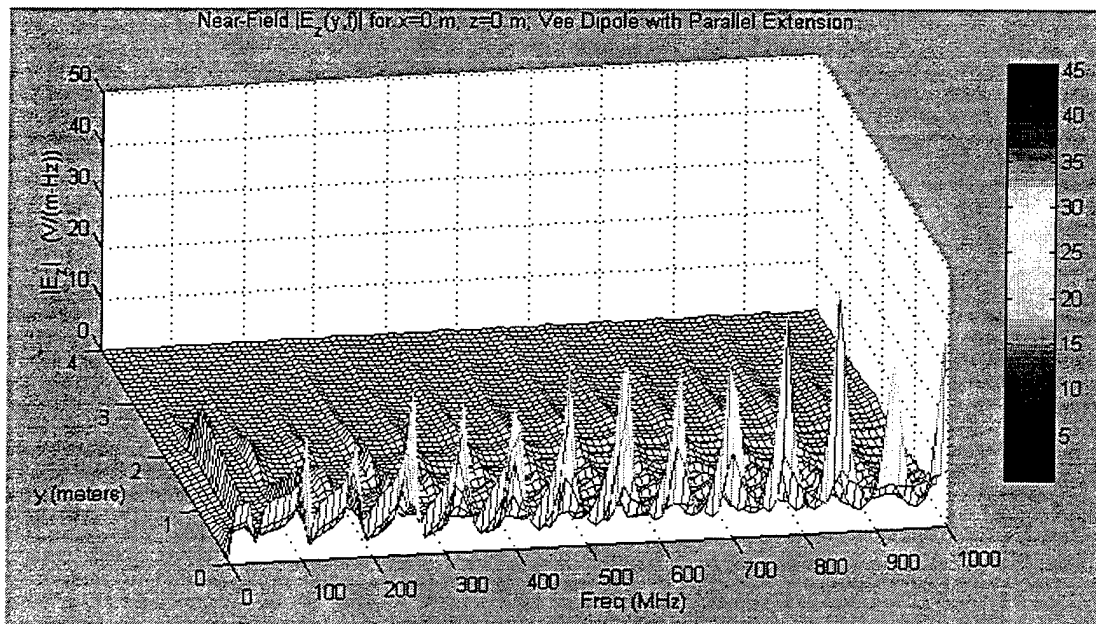


Figure 44. The Frequency-Domain E-Field for the Vee Dipole with Parallel Extension

Additionally, Figure 45 confirms the periodicity of the voltage standing wave for each frequency, just as was seen for the Vee Dipole in Figure 41. Also, the coupling efficiency for every frequency above the limit defined by $f = c/L$, where c is the speed of light, can also be determined from the magnitude of the E-field within the limits of the antenna structure.

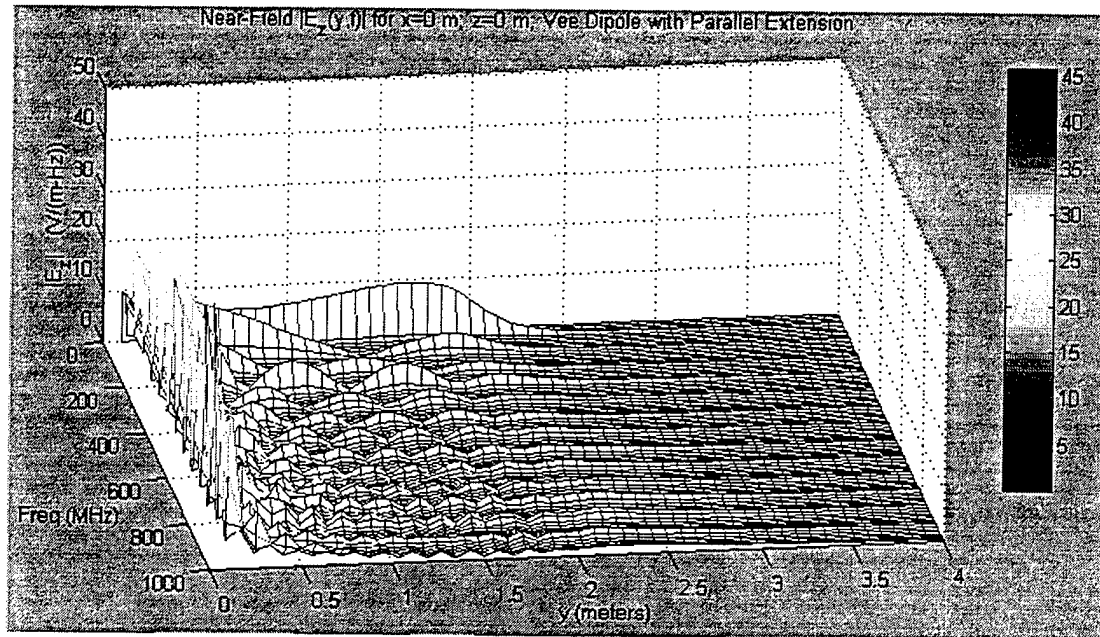


Figure 45. A Different Perspective of the Plot Shown in Figure 44

The time-domain representation of the E-field for this antenna, Figure 46, clearly shows the initial impulse propagating out of the antenna into free space. Like the Vee Dipole, this, too, has a more directive nature than a simple dipole; however, in contrast to the earlier antennas, this one's propagating wavefront is not at all semicircular. It is clear that the wavefront begins to flatten out and approach a planar front as it is forced to propagate further down a transmission line. Also, though not directly evident in this frame, but exceptionally clear in the movie rendition of the time-domain fields, the reflection coefficients at each end of the antenna do, indeed, match the theoretical predictions.

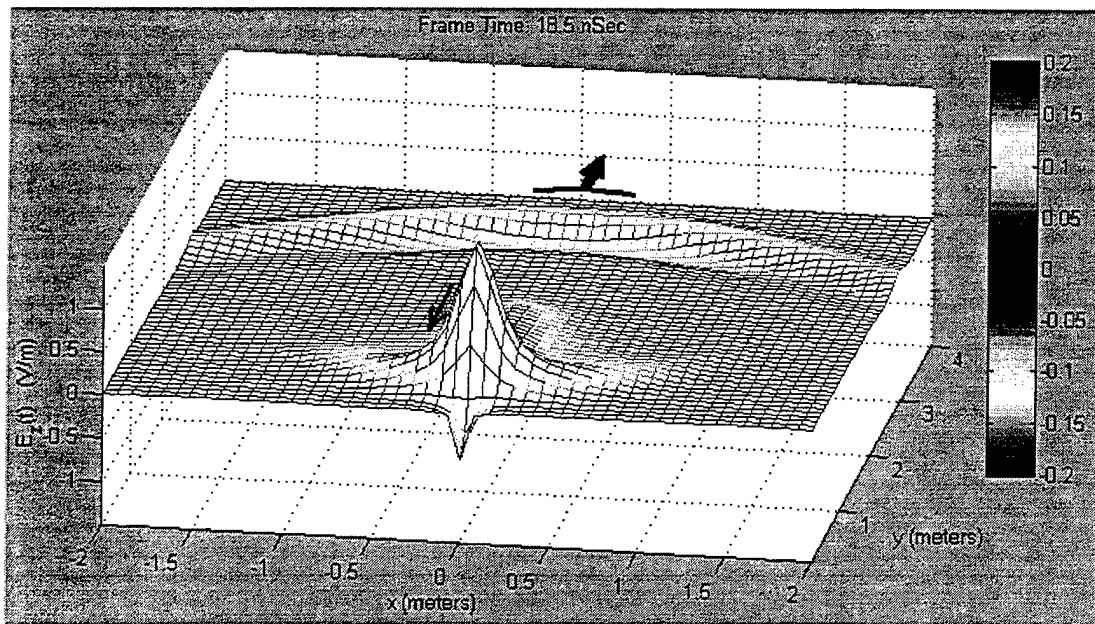
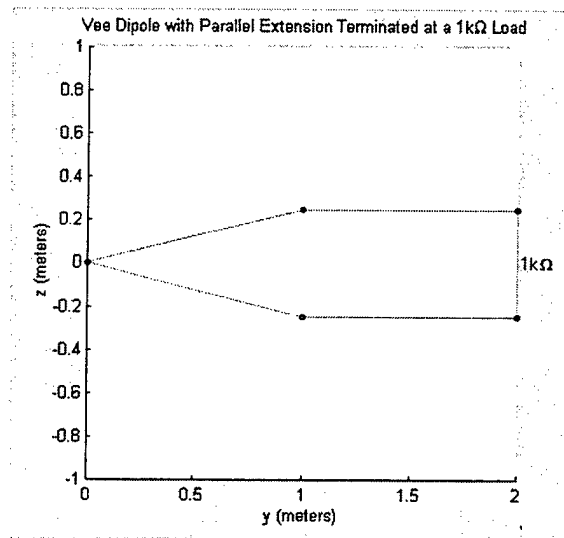


Figure 46. The Time-Domain E-Field of the Vee Dipole with Parallel Extension

In order to eliminate, or at least reduce, the amount of energy reflected back to the feed from the aperture, a resistance would need to be added along the parallel section of this antenna. The resistance would serve to convert the energy from the uncoupled resonant frequency waves into heat, and thereby attenuate the reflected current each time it would pass through the resistance. This is done in the exact same manner as was discussed for the TEM Horn in Chapter II. In order to test this and additional transmission line phenomenology with the software, one final antenna structure was analyzed.

4. The Vee Dipole with a Parallel Extension Terminated at a 1 k Ω Load

The final two-wire structure tested utilizing the analysis software was the Vee Dipole with the parallel extension now terminated at a 1k Ω load. As noted in Figure 47, this antenna differs from that last considered in only the addition of the vertical wire load segment at the aperture. The purpose of this modification was to ensure that the software would verify a change in both the reflection coefficient at the aperture and the attenuation of the reflected energy caused by uncoupled low frequency waves.



**Figure 47. Vee Dipole with Extension
Terminated at a $1k\Omega$ Load**

It can be shown that for a transmission line of this type, the reflection coefficient at the aperture end will approach -1 for those frequencies that do not produce coupling [9]. In addition, the fact that the load is along a wire segment that is 0.5 meters long in the vertical axis, means the loaded end should exhibit some radiation properties separate from the rest of the transmission line.

Looking at Figure 48, the frequency-domain representation of the E-field for this particular antenna structure reveals a flatter frequency response of the antenna across all frequencies. This is very different from the “spikey” nature of the frequency response seen in the past antennas’ plots (Figures 40 and 44), and implies the load is, in fact, attenuating the uncoupled resonant frequency waves. There are certainly still peaks along the frequency band; however, the nulls are not as deep and the overall magnitude of the E-field is more constant across the entire band. Throughout the antenna length, the fields observed at the lower frequencies have been attenuated, and now more closely match the values for the frequencies able to couple and launch into free space. This does not mean that the resistive load will allow more frequencies to couple than for the unloaded antenna. Rather, it is merely attenuating the reflected energy at those uncoupled

frequencies. If any additional frequencies are able to couple more efficiently, it is most likely due to the 0.5 meter segment that was added to support the load. As stated earlier, this additional segment, although loaded, will not only add length to the original antenna, thereby improving the low-frequency characteristics, but also serve as an additional focused radiation point for the antenna. This additional radiation point by no means aids the radiation performance of the antenna; rather, it merely serves as another source point on the antenna structure.

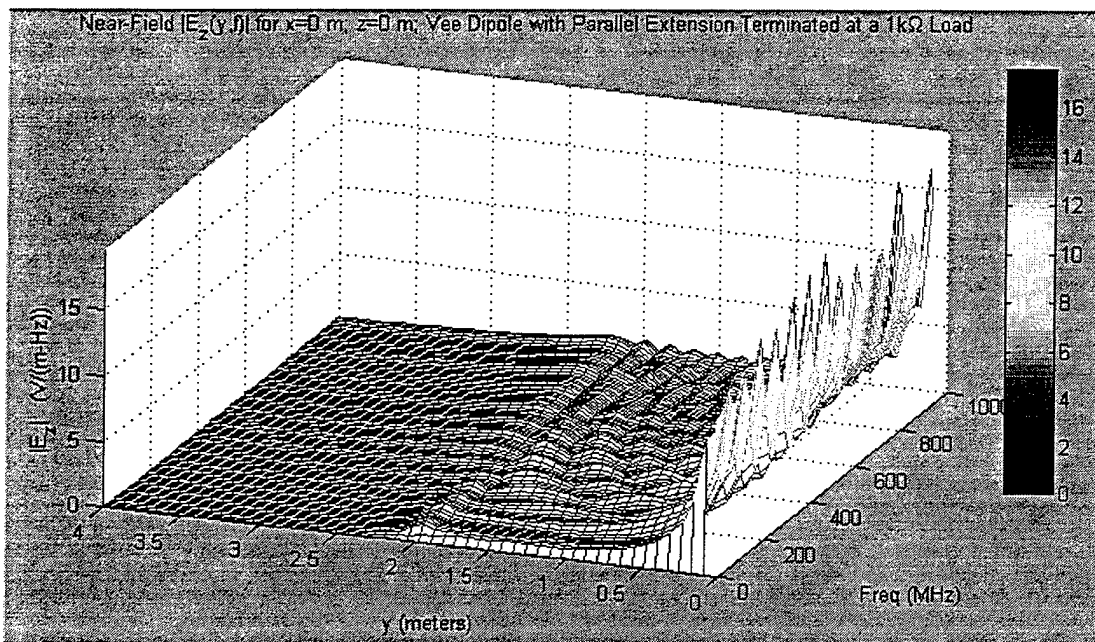


Figure 48. Frequency-Domain E-Field of the Extended Vee Dipole Terminated at a 1kΩ Load

The final area of interest is the 2-meter point in Figure 48. Specifically, this is where the aperture was replaced by the vertical loaded segment. Across all frequencies, there appears a ridge of slightly higher field strength. This is undoubtedly due to the radiant nature of the loaded segment. On one hand it is dissipating low-frequency energy, but on the other, it is most certainly serving as a source point. If this is true, then there should be evidence of this in the time-domain plots.

Figure 49 is a frame selected from the time-domain movie of this antenna's E-field. Chosen at a point after the initial pulse has reached the loaded end and is reflecting back to the feed point, this plot graphically depicts the predicted propagation pattern. Specifically, it should be noted that the wave propagating in the positive y -direction displays the same characteristics of the initial wave propagating through the antenna of Section 5.A.3, as depicted in Figure 46. In addition, the reflected wave and part of the leading edge of the initial wavefront are displaying propagation characteristics similar to those of the simple dipole. Also, the portion of the reflected wave propagating along the wire structure is definitely at a lower magnitude than the reflected waves from either of the last two antenna models. In hindsight, then, this antenna has functioned in many ways as the superposition of two different antennas upon each other. Although this was not directly predicted through theory, the end result was, at a minimum, enlightening.

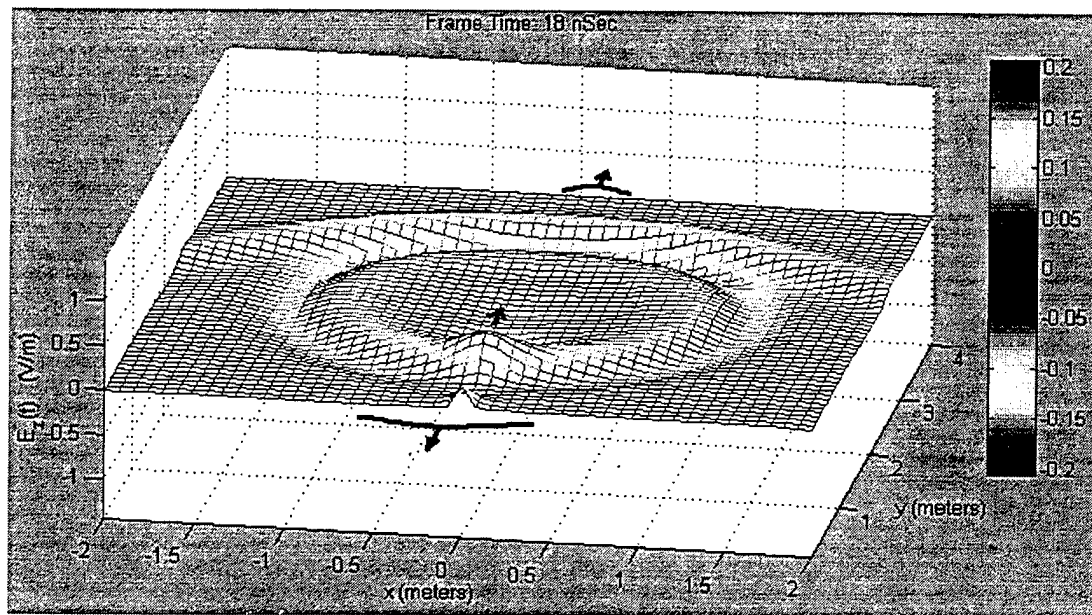


Figure 49. Time-Domain E-Field for the Extended Vee Dipole Terminated at a $1\text{k}\Omega$ Load

It must be stated that the software not only was able to visually confirm that behavior which was theoretically predicted, but also provided heuristic insight. Overall, it is easy to see that the graphical analysis, while perhaps not as specific as a methodical

mathematical process, does, indeed, allow the user to display certain antenna performance characteristics, and further present an accurate picture of the near-fields for further study and in-depth analysis.

B. TEM HORN PROTOTYPE DESIGN

Having validated the software's capabilities as an analysis tool, we now turn our attention to an antenna for which the software was initially designed to support, the TEM Horn. Figure 50 is the wire model used to emulate the prototype TEM Horn. Its dimensions are 1.8288m (6ft) long, 0.1524m (6in) high, and 0.1026m (4in) deep. As discussed in Chapter II, resistive loading has been added along the parallel extensions. Resistive loads are placed between each of the dark nodes on the right half of the antenna. Figure 50 is a wire-based analog of the continuous surface structure shown in Figure 1. As such, it offers current paths which are the same as the actual antenna. Extensive validations [5] have shown that under the conditions observed here, such wire structures provide accurate emulations of solid surface antennas.

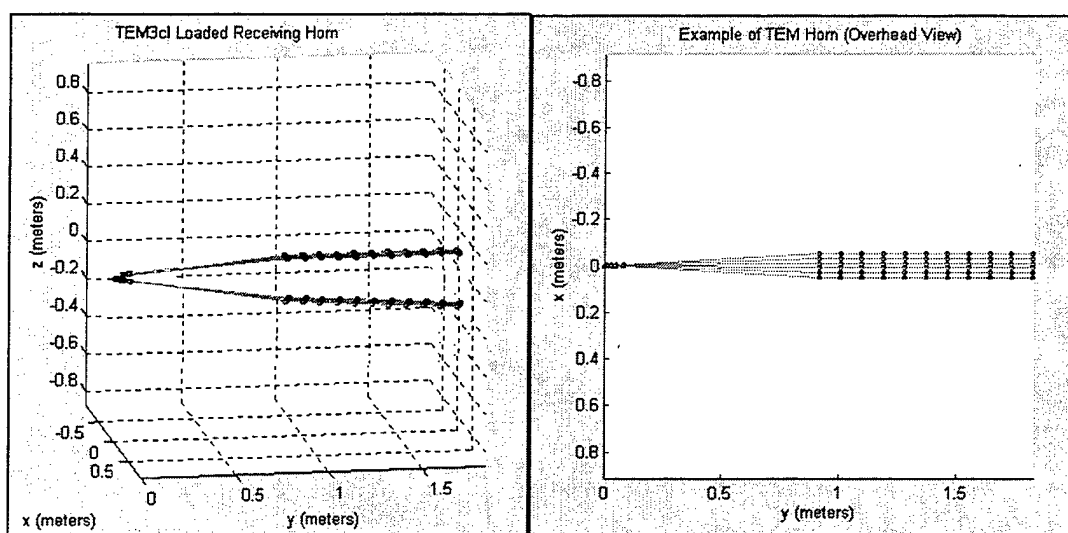


Figure 50. 3-D and Overhead Views of the Prototype TEM Horn

The goal of the analysis for this antenna, then, is first to develop an understanding of the E-field generated within the near field by the TEM Horn, and second, to demonstrate that the prototype TEM Horn either meets or fails to meet the design criteria of an UWB impulse receiving antenna operating in the 10MHz-10GHz bandwidth. To do so, we will analyze the frequency-domain E-field plot and garner whatever insight we may prior to investigating the time-domain response of the antenna. The analysis of the E-fields in the frequency and time domains should provide us with enough visual evidence of the antenna's success in meeting the design criteria set forth earlier.

The first concern for the TEM Horn is a need for a relatively flat response across all frequencies. For this antenna, an initial frequency range of 0 to 2 GHz was analyzed. Figure 51 depicts the TEM Horn's E-field as a function of frequency and distance along the antenna's centerline axis, and Figure 52 provides the E-field at the antenna feed point, only. What should be evident immediately is that the response of the antenna, while not perfectly straight across the top of the plot, provides very little deviation from one frequency to the next. The "ridge" present at a very low frequency may indicate a

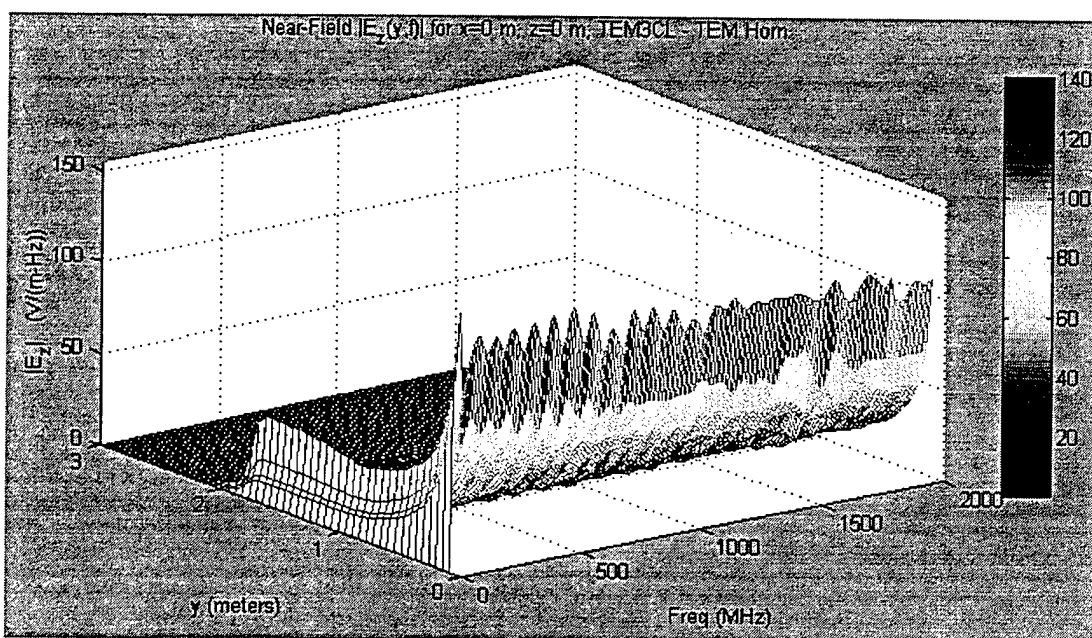


Figure 51. Frequency-Domain Plot of the E-Field for the Prototype TEM Horn

problem in the design; however, this plot does not provide a small enough scale to accurately identify a specific frequency. Although we expect to have poor response below the lower design limit of 10 MHz, we must ensure the response from 10-100 MHz is not causing fluctuations in the field, thereby causing the output to be distorted and lack fidelity. Thus, the 0-100 MHz frequency range will be investigated in more depth later.

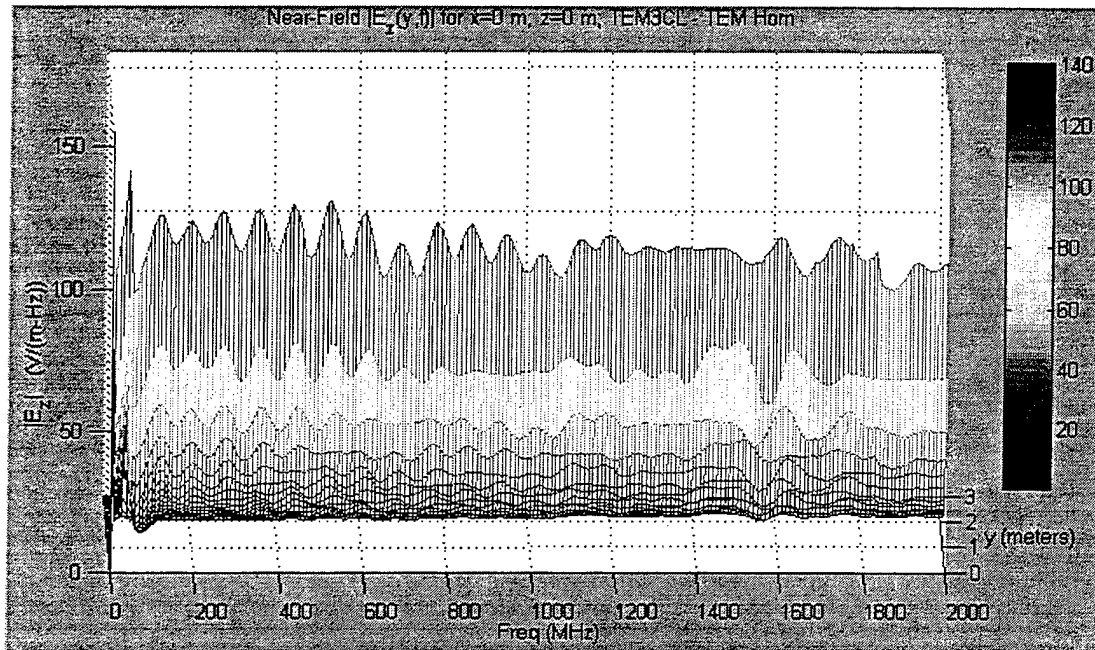


Figure 52. Frequency Response of the TEM Horn at the Feed Point

Looking at Figure 53, we now are able to view the response of the antenna vs. the distance along the centerline axis. Clearly, the field strength along the y -axis weakens rapidly yet smoothly for all frequencies above 200 MHz. Below this point, we notice a “bulge” beginning to form at the end of the antenna for the lower frequencies. This bulge, circled in white, is an indication that some of this low frequency energy is not completely coupled and launched into free space. As a result, the antenna may have degraded performance (i.e., less-than-desirable fidelity) at the very low end. To investigate this further, a low-frequency analysis is a must.

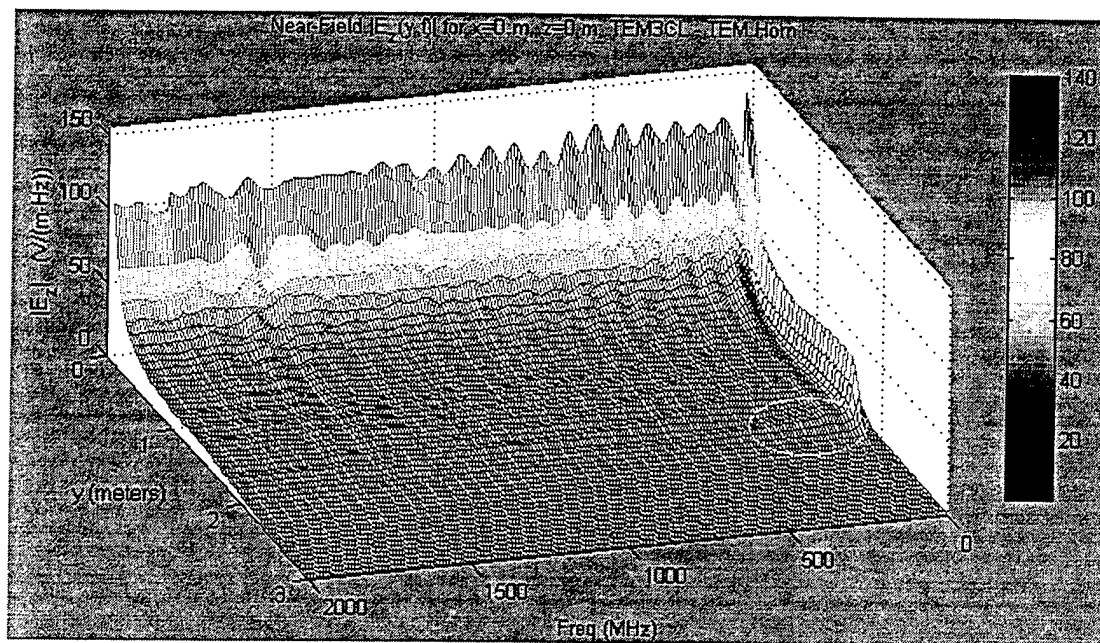


Figure 53. Frequency-Domain Plot of the TEM Horn's E-Field with Area of Interest Highlighted

Prior to beginning the low frequency analysis, though, a study of the time-domain waveform was conducted to verify the assumptions made above. Figures 54 and 55 are frames taken from the time-domain propagation movie. At 9nsec after application of the Gaussian impulse, the waveform in Figure 54 is maintaining excellent shape as it propagates through the antenna, with no noticeable deformation occurring.

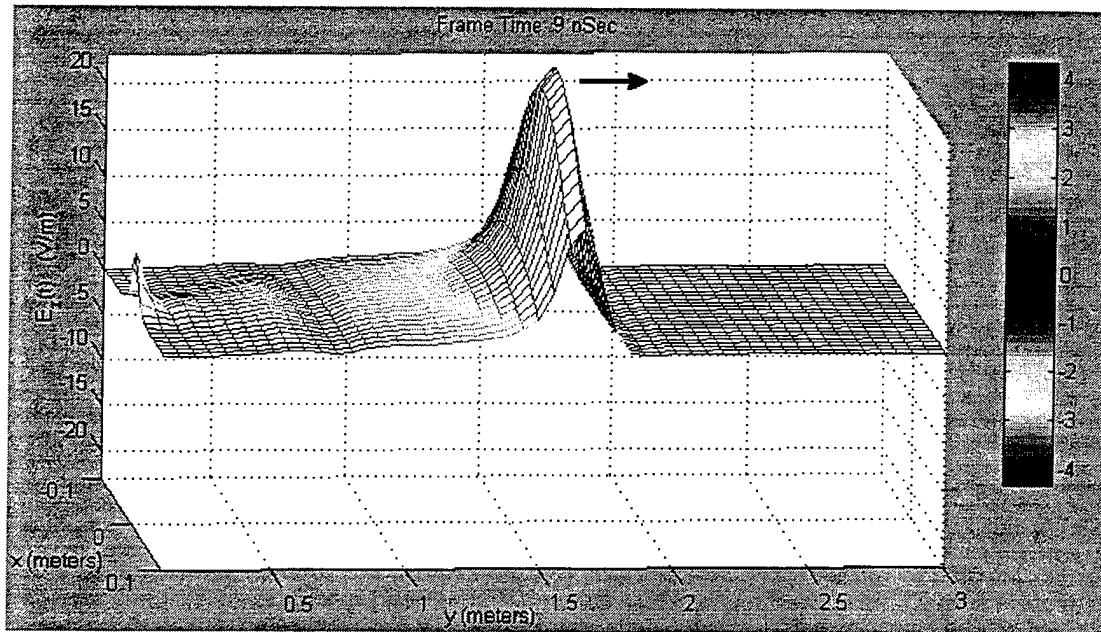


Figure 54. Time-Domain E-Field for the Prototype TEM Horn

At 10nsec after application of the Gaussian impulse, the wavefront reaches the aperture, and as seen in Figure 55, most of the energy launches into free space. This wave, a double Gaussian, is approximately 2.5m from the antenna feed point. The low frequency energy that did not couple is then reflected back through the resistive elements and dissipated as heat. The large bulge between 1.25m and 1.8m is most of the reflected wave being “trapped” in the resistive load and dissipated. The strong field near the feed (at $y=0.1\text{m}$) is of no concern. While it may appear to be significant in this plot, one must remember that the magnitude of the E-field is dependent upon the area within which it is concentrated. Beyond the aperture, the area is growing rapidly, since the wave is in free space; whereas, at the feed, the field is confined within the few centimeters separating the top and bottom plates. Thus, the E-field at the antenna feed point, while small, appears amplified because it is so confined.

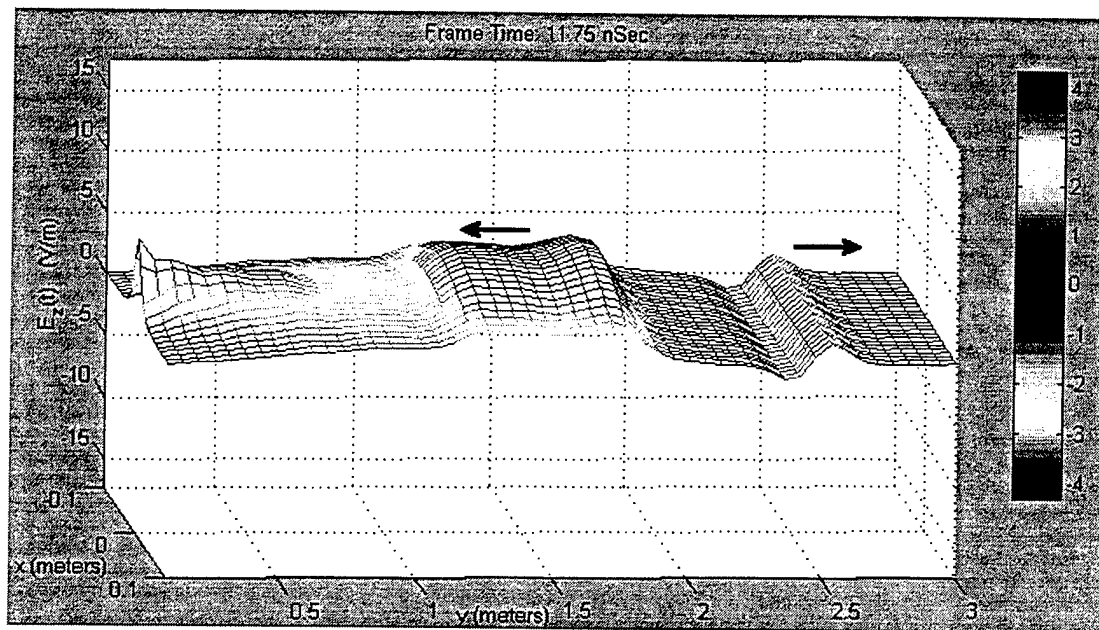


Figure 55. Time-Domain E-Field for the Prototype TEM Horn

The reflection at the aperture is still cause for concern, since it validates the prediction made after viewing the antenna's frequency response. To determine the exact frequencies that are causing this, the antenna was next analyzed over the frequency range 0 to 100 MHz. Figures 56 through 58 are plots of the antenna's frequency response over this range. What is immediately noticeable is the ridge of higher-than-normal field strength located at approximately 40 MHz, which is well within the antenna's required bandwidth. Unfortunately, neither the frequency- nor time-domain plots give any indication as to the cause of this anomaly in the frequency response. Specifically, the answer can only be determined through in-depth analysis of the antenna's key parameters at the frequency in question. While the software in this case may lack the depth to determine the cause for poor frequency response of an antenna, it does have the range and ability to at least display possible areas of concern for the designer.

In Chapter II, it was determined the sensitivity of the prototype TEM horn could suffer by as much as 3 dB near 26 MHz. From Figures 56 through 58, it is clear that the frequency response of the prototype TEM horn experiences a maximum deviation of only 1.5 dB in the low frequency range, which will have a negligible impact on performance.

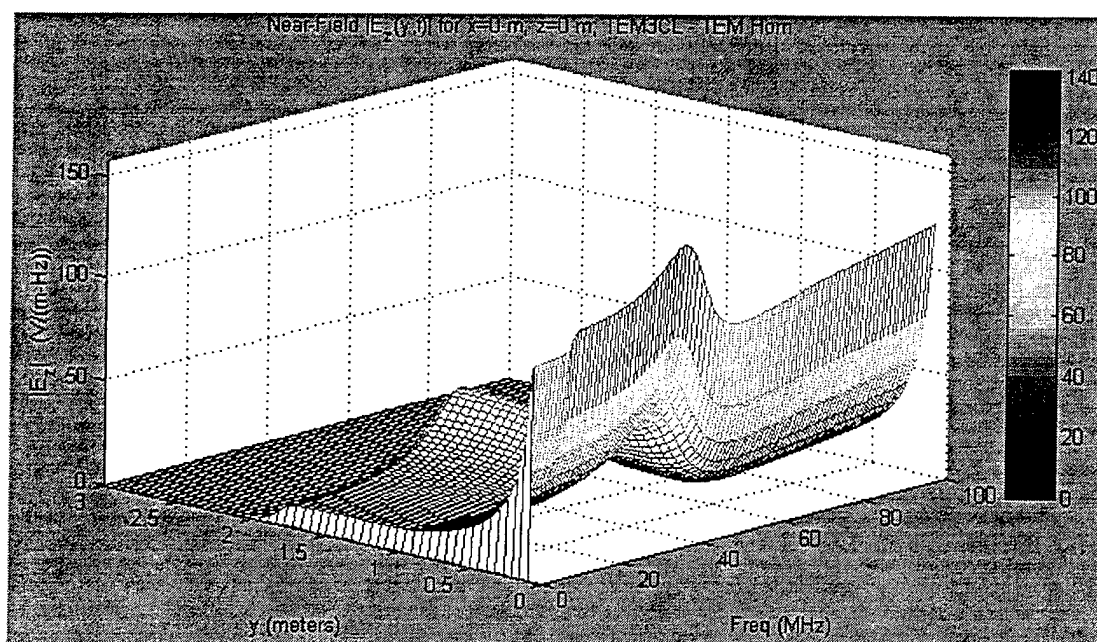


Figure 56. Frequency-Domain E-Field for the TEM Horn at Low Frequencies

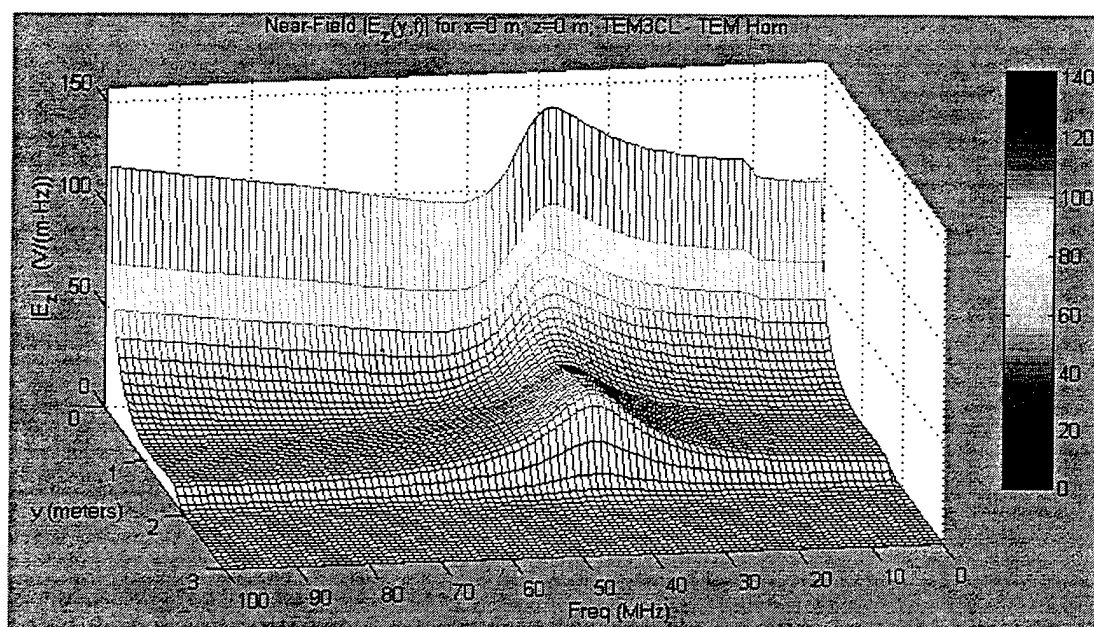


Figure 57. Prototype TEM Horn E-Field Showing Possible Resonant Frequency

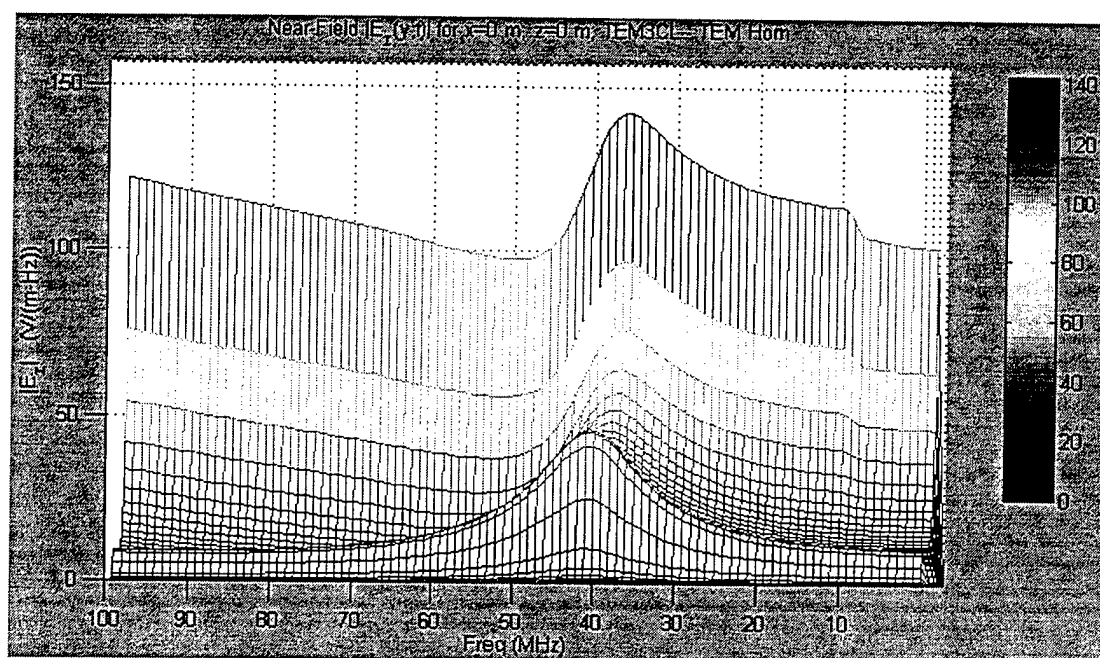


Figure 58. Close-up of Possible Resonant Frequency

As a final analysis step, the TEM horn was also set up as a receiving antenna with a double-Gaussian impulse functioning as the incident field. Recall that reciprocity dictates that the radiation pattern of an antenna is the same whether it is used as a transmitting antenna or receiving antenna. Although the excitation for our transmitting and receiving cases is not exactly the same, it makes sense that the general behavior of the TEM horn's near fields will be similar. In the ideal case, however, further validation of the software's accuracy could then be accomplished by matching the two near fields derived under the two different circumstances if the excitation in both cases were the same (i.e., the transfer impedances were equal).

In Figure 59, the double-Gaussian incident impulse is seen propagating towards the TEM horn. At $t=13.5\text{nsec}$, the incident impulse field impacts the antenna feed, and a reflected wave begins to form and propagate in the $+y$ direction along the antenna's centerline axis. Figure 60 clearly illustrates the formation of the reflected wave at $t=15\text{nsec}$. Finally, at $t=16.5\text{nsec}$, the reflected wavefront reaches the aperture, and as

seen in Figure 61, the resultant transmitted and reflected fields behave and look the same as they did in the transmitting antenna case (Figure 55).

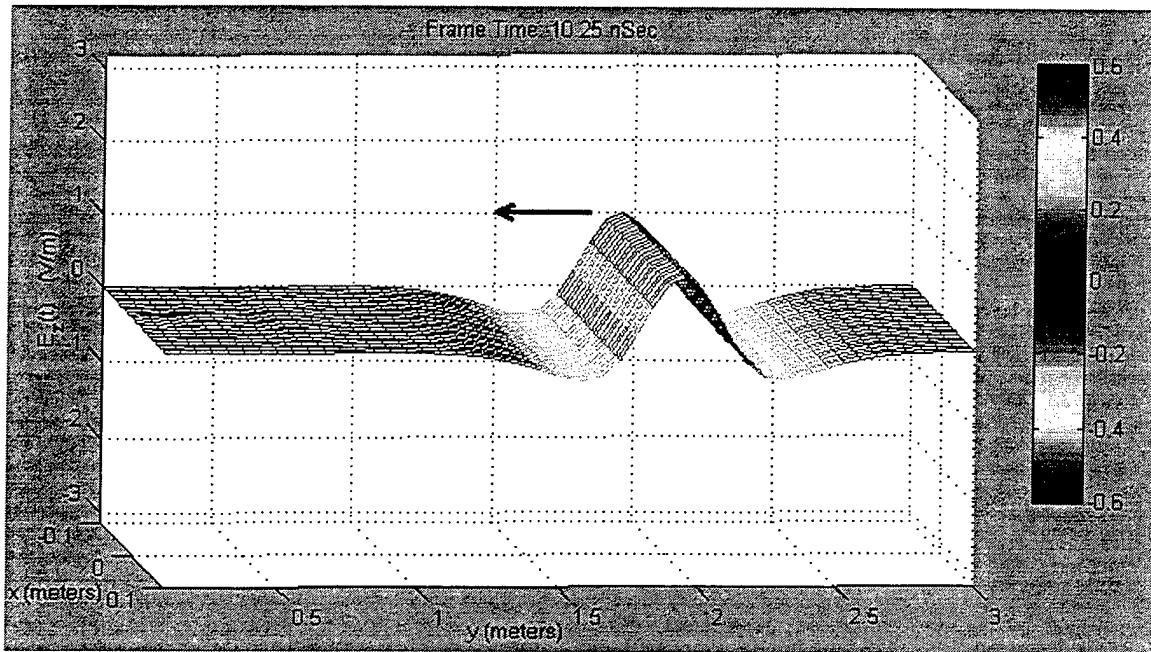


Figure 59. Incident Double-Gaussian Impulse E-Field

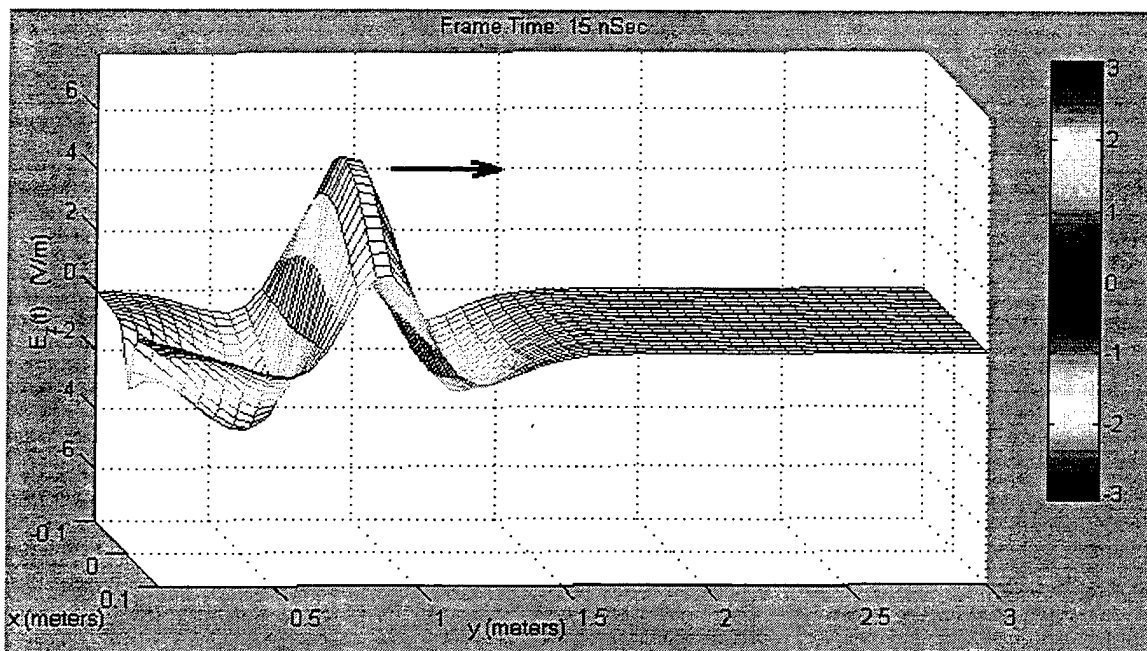


Figure 60. Time-Domain Reflected E-Field within the TEM Horn Receiver

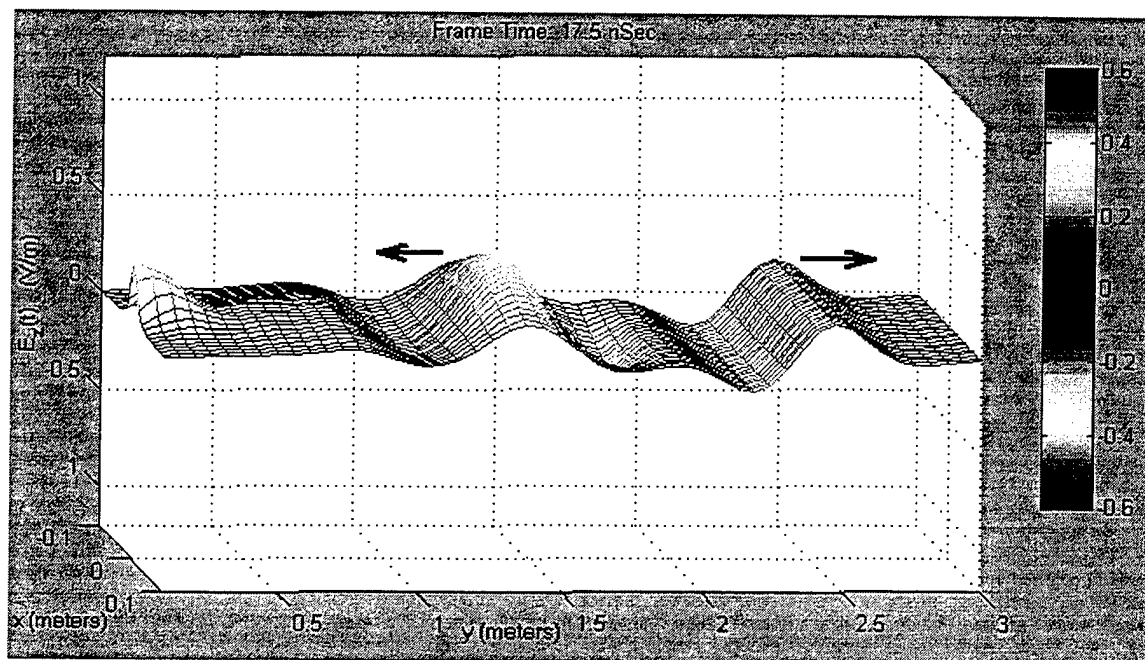


Figure 61. Resultant Transmitted and Reflected Fields at the TEM Horn Aperture

VI. CONCLUSIONS

The focus of effort of this research was in the creation of a high-level graphics package that would primarily aid in the design and analysis of an UWB antenna and secondarily function as an instructional aid in the understanding of basic radar and transmission line theory. In an effort to reach these goals, the MATLAB function *NFLDVIS.M* was written and designed to work hand-in-hand with the data output files created through *AWAS*. Specifically, with a general knowledge of both *AWAS* and *NFLDVIS.M*, a user could build a wire model of an antenna or transmission line and then actually view the magnitude of the associated Near Fields as either a function of frequency and distance, or time and distance.

As a design tool, the software was demonstrated to perform well with both simple, classical antenna designs, as well as state-of-the-art TEM Horns. Although the designer does require the use of two separate sets of software, the synchronization between the two should make that issue somewhat transparent. *NFLDVIS.M* is designed to be user-friendly, and with the designer only required to use a keyboard for typing in the title for his plots, this goal has been met.

As for the TEM Horn itself, the software indicated the prototype antenna design may have problems at the low frequency end of the spectrum. The one anomaly present at approximately 40 MHz may be indicative of a natural resonance in the antenna, but since the antenna tested in *AWAS* (Figure 50) is not exact in its representation of the resistive loading design, it is probable the reflection noted in Figures 56 through 58 will be attenuated more aggressively by the logarithmic loading scheme planned. In addition, recall that in Chapter II, it was theorized that the antenna may suffer a sensitivity degradation by as much as 3 dB near 26 MHz and 6 dB near 13 MHz; however, there was no indication of such in the frequency response curves created by the software. On the contrary, the fluctuation in the magnitude of the frequency response of the antenna never exceeded 1.5 dB over the entire tested range. To absolutely guarantee against any

degradation in the sensitivity, increasing the length, L , of the TEM horn to 4.775m will keep the lower frequency limit outside of the intended spectrum of the TEM horn.

It was further theorized in Chapter II that the TM_{02} mode would be excited at some frequency above 6.225 GHz, thus causing distortion of the output waveform. Unfortunately, due to the limitations of *AWAS* discussed in Chapter III, the exact frequency where the TM_{02} mode was excited could not be verified. Assuming that were the case, though, the TEM horn would require a redesign to guarantee distortionless transmission throughout the entire frequency band. Specifically, if the antenna's total aperture height was reduced to 94 mm, then $f_{TM_{02}}$ will definitely occur outside the frequency band of interest.

At no time did the software package ever produce faulty data or display unjustifiable renderings, except for the occasional "garbage-in, garbage-out" operator errors. In other words, every plot or time-domain movie the software generated was completely in line with the theoretical expectations of the particular antenna or transmission line in question. In the long run, though, the true future for the *NFLDVIS.M* function may lie in its ability to provide a detailed glimpse into what most people never see. Beginning with the verification of the software's accuracy, it became increasingly evident that visualization of the near fields in either the frequency- or time-domain proved an invaluable educational tool. For even though one may hear or read about concepts, or take measurements to confirm theory, the actual ability to see not only the effects, but also the cause adds a new, logical dimension to the learning process. The old adage, "a picture is worth a thousand words" rings true. But even this software cannot tell you everything. As seen in Section 5.B, it is capable of indicating potential problem areas in a design, but it can never tell you the "why". That answer is left for the designer.

LIST OF REFERENCES

1. A. V Oppenheim and A. S. Willsky, *Signals and Systems*, Prentice Hall, New Jersey, 1983.
2. W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*, John Wiley & Sons, New York, 1981.
3. A. R. Ondrejka, J. M. Ladbury, and H. W. Medley, "TEM Horn Antenna Design Guide," *National Institute of Standards and Technology*, unpublished report.
4. R. C. Robertson and M. A. Morgan, "Ultra-Wideband Impulse Antenna Study and Prototype," *Naval Postgraduate School Final Report*, NPSEC-93-010, March 1993.
5. A. R. Djordjevic et al., *Analysis of Wire Antennas and Scatterers: Software and User's Manual*, Artech House, Boston, 1990.
6. D. V. Adamiak, *Antenna Near-Field Visualization Software User's Manual*, unpublished.
7. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, New Jersey, 1996.
8. R. S. Elliott, *Antenna Theory and Design*, Prentice Hall, New Jersey, 1981.
9. D. K. Cheng, *Field and Wave Electromagnetics*, Addison-Wesley, Massachusetts, 1992.

APPENDIX A. NEAR-FIELD VISUALIZATION SOFTWARE (NFLDVIS.M)

```
function nflldvis(action)
%NFLDVIS This Graphical-User-Interface (GUI) provides the antenna designer
% with a tool to view the electric and magnetic fields generated
% within and in close proximity to an Ultra-Wideband Impulse Antenna.
% This program uses AWAS-generated near-field (*.nfl) and geometry
% (*.geo) files as input, and provides graphical output of the
% near-fields in both the frequency and time domains. In the
% frequency domain, the designer is able to view the fields in 3-D
% as a function of frequency along the antenna axis. In the time
% domain, a movie is created, so that the designer may view the
% antenna as it either transmits or receives an impulse waveform.
% For the case of the antenna operating as a transmitter, the user
% may view the wave as it propagates through the antenna and is
% "launched" into free space and/or reflected back to the feed point.
% For the case of the antenna operating as a receiver, the user may
% view the wave as it propagates through free space, impacts the
% antenna aperture, and reflects off the antenna or continues to
% propagate within the antenna structure. As an additional option,
% the designer may simply playback movies that were created earlier
% and saved as .mat files.
%
% Follow the prompts provided in the text window at the top of the
% figure to step through the analysis process. The "slides" at the
% bottom of the figure window allow the experienced user to control
% the display of the data. Default settings are already built into
% the program. The program also prompts the user when input from a
% slide, popup menu, or text box is required.
%
% Display of the frequency-domain fields is controlled by the "Plot"
% menu on the menubar at the top of the window, while display of the
% time-domain representations is controlled by the "Movie" pushbutton
% on the right side of the window.
%
% A full demonstration of the software is provided merely by pressing
% the "Demo" pushbutton on the right side of the window.
%
% Although a "Print" menu is available on the menubar at the top of
% the figure window, the only menu selection which is currently
% operable is the "Save..." option, which allows the user to save the
% entire figure as a .TIF file. The other options are currently
% reserved for future use. When operational, they will allow for
% printing of either the main axes separately or both the main axes
% and the colorbar together. Since MATLAB does not yet offer printing
% of select "frames" (axes) within a figure window, these two options
```



```

%      have been deactivated, awaiting a future release of MATLAB. If a
%      print of the entire GUI (ie, buttons, graph, colorbar, and all other
%      uicontrols) is desired, merely use the normal "Print" submenu found
%      under the "File" menu at the top left corner of the figure window.

% Written & designed by Dave Adamiak
% Copyright (c) 1997
% Last mod: 10/10/98

% Set the required initial conditions & create the GUI:
if nargin<1,
    action='initialize';
end;

switch(action)
%%
case 'initialize'
    oldFigNumber=watchon;

    figNumber=figure( 'Color',[0.6 0.6 0.6], ...
        'Name','Transient Field Visualization for Ultra-Wideband Antenna Design   by Dave
Adamiak', ...
        'NumberTitle','off', ...
        'Units','normalized','Position',[0 0.036 1 0.9], ...
        'Visible','off', ...
        'BackingStore','off');
    axHnd1=axes('Color',[1 1 1], 'Position',[0.075 0.2 0.675 0.6], ...
        'Visible','off','Tag','Axes1');

    text(0,0.1,'Press the "Start" button to begin the analysis,', ...
        'HorizontalAlignment','center');
    text(0,0,'Press the "Demo" button to see a demonstration of the software,', ...
        'HorizontalAlignment','center');
    text(0,-0.1,'Press the "Movie" button to replay a saved movie or create a new one,', ...
        'HorizontalAlignment','center');
    text(0,-0.2,'Or click on the "Plot" menubar to create various plots using data from your last
session.', ...
        'HorizontalAlignment','center');
    text(0,-0.4,'All commands will be issued in the text window at the top of this screen.', ...
        'HorizontalAlignment','center');
    axis([-1 1 -1 1])

%=====

```

```

% The INFO button
callbackStr = 'nflldvis("info");
infoHndl = uicontrol('Units','normalized', ...
    'Position',[0.795 0.8168498168498168 0.08 0.07875457875457875], ...
    'String','Info', 'Callback',callbackStr, 'Tag','Pushbutton1');

%=====
% The START button
callbackStr = 'nflldvis("start");
startHndl = uicontrol('Units','normalized', ...
    'Position',[0.89125 0.8168498168498168 0.08 0.07875457875457875], ...
    'String','Start', 'Callback',callbackStr, 'Interruptible','on', ...
    'Tag','Pushbutton2');

%=====
% The DEMO button
callbackStr = 'nflldvis("demo");
demoHndl = uicontrol('Units','normalized', ...
    'Position',[0.89125 0.7142857142857143 0.08 0.07875457875457875], ...
    'String','Demo', 'Callback',callbackStr, 'Tag','Pushbutton3');

%=====
% The MOVIE button
callbackStr = 'nflldvis("movie");
movieHndl = uicontrol('Units','normalized', ...
    'Position',[0.89125 0.6105006105006106 0.08 0.07875457875457875], ...
    'String','Movie', 'Callback',callbackStr, 'Tag','Pushbutton4');

%=====
% The CONTINUE button
continuHndl = uicontrol('Units','normalized', ...
    'Position',[0.89125 0.5073260073260073 0.08 0.07875457875457875], ...
    'String','Continue', 'Enable','off', 'Tag','Pushbutton5');

%=====
% The RESET button
callbackStr = 'nflldvis("reset");
resetHndl = uicontrol('Units','normalized', ...
    'Position',[0.89125 0.4047619047619048 0.08 0.07875457875457875], ...
    'String','Reset', 'Callback',callbackStr, ...
    'Tag','Pushbutton6');

%=====
% The CLOSE button
callbackStr = 'close(gcf);
closeHndl = uicontrol('Units','normalized', ...

```

```

'Position',[0.89125 0.3003663003663004 0.08 0.07875457875457875], ...
'String','Close', 'Callback',callbackStr, 'Tag','Pushbutton7');

%=====
% The Instructional Text Box
edittextHndl = uicontrol('Units','normalized', 'BackgroundColor',[1 1 1], ...
    'Position',[0.09875 0.9065934065934066 0.60375 0.0695970695970696], ...
    'FontSize',[10], 'Max',[2], 'Style','edit', 'String', ...
    'Look Here for All Commands & Prompts!', 'Tag','EditText1');

%=====
% The Near-Field Group popup menu
popupStr1 = ('Near-Field Group');
grppopupHndl = uicontrol('Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.09875 0.8589743589743589 0.12625 0.0347985347985348], ...
    'String',popupStr1, 'Enable','off', 'Style','popupmenu', 'Tag', ...
    'PopupMenu1', 'Value',1);

%=====
% The Total Frames popup menu
popupStr2 = ('Total Frames');
frampopupHndl = uicontrol('Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.3375 0.8589743589743589 0.12625 0.0347985347985348], ...
    'String',popupStr2, 'Enable','off', 'Style','popupmenu', 'Tag', ...
    'PopupMenu2', 'Value',1);

%=====
% The Frames-Per-Second popup menu
for n=1:5, fpsstr(n,:)=sprintf('%13.0f,2*n); end
popupStr3 = char('Frames-per-sec',fpsstr);
fppopupHndl = uicontrol('Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.57625 0.8589743589743589 0.12625 0.0347985347985348], ...
    'String',popupStr3, 'Enable','off', 'Style','popupmenu', 'Tag', ...
    'PopupMenu3', 'Value',1);

%=====
% The Z-Axis Scaling slider & its associated text boxes
zscalslidHndl = uicontrol('Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'Position',[0.1 0.06227106227106227 0.1475 0.03663003663003663], ...
    'Style','slider', 'Enable','off', 'Tag','Slider1');
zscaltextHndl1 = uicontrol('Units','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...

```

```

'Position',[0.11875 0.102564103 0.11 0.024420024], ...
'String','Z-Axis Scaling', 'Style','text', 'Tag','StaticText1');
zscaltextHndl2 = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.116666666667 0.0347985347985348 0.0175 0.02564102564102564], ...
'String','0', 'Style','text', 'Tag','StaticText2');
zscaltextHndl3 = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.215 0.0347985347985348 0.0175 0.02564102564102564], ...
'String','1', 'Style','text', 'Tag','StaticText3');

%=====
% The Azimuth View Editable Text Box
azieditHndl = uicontrol('Units','normalized', 'BackgroundColor',[1 1 1], ...
'Position',[0.3266666666666667 0.06227106227106227 0.05 0.03663003663003663], ...
'String','-37.5', 'Enable','off', 'Style','edit', 'Tag','EditText2');
azitextHndl = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.3229166666666667 0.102564103 0.0575 0.02564102564102564], ...
'String','Azimuth', 'Style','text', 'Tag','StaticText4');

%=====
% The Elevation View Editable Text Box
eleditHndl = uicontrol('Units','normalized', 'BackgroundColor',[1 1 1], ...
'Position',[0.4191666666666666 0.06227106227106227 0.05 0.03663003663003663], ...
'String','30', 'Enable','off', 'Style','edit', 'Tag','EditText3');
eltextHndl = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.4154166666666667 0.102564103 0.0575 0.024420024], ...
'String','Elevation', 'Style','text', 'Tag','StaticText5');

%=====
% The Color-Axis Scaling slider & its associated text boxes
cscalslidHndl = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.555 0.06227106227106227 0.1475 0.03663003663003663], ...
'Style','slider', 'Enable','off', 'Tag','Slider2');
cscaltextHndl1 = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.5725 0.102564103 0.1125 0.024420024], ...
'String','Color-Axis Scaling', 'Style','text', 'Tag','StaticText6');
cscaltextHndl2 = uicontrol('Units','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'Position',[0.57125 0.0347985347985348 0.0175 0.02564102564102564], ...
'String','0', 'Style','text', 'Tag','StaticText7');
cscaltextHndl3 = uicontrol('Units','normalized', ...

```

```

        'BackgroundColor',[0.752941 0.752941 0.752941], ...
        'Position',[0.67 0.0347985347985348 0.0175 0.02564102564102564], ...
        'String','l', 'Style','text', 'Tag','StaticText8');

%=====
% The Plot Options menu & sub-menus
plotHndl = uimenu('Label','Plot', ...
    'Tag','PlotMenu');
plotimpuls = uimenu('Parent',plotHndl, ...
    'Label','Frequency-Domain Fields', 'Tag','Plotuimenu1');
plotEy1 = uimenu('Parent',plotimpuls, 'Callback','nflldvis("plotEy1")', ...
    'Label','Ey(f)', 'Tag','Plotuimenu2');
plotEz1 = uimenu('Parent',plotimpuls, 'Callback','nflldvis("plotEz1")', ...
    'Label','Ez(f)', 'Tag','Plotuimenu3');
plotHx1 = uimenu('Parent',plotimpuls, 'Callback','nflldvis("plotHx1")', ...
    'Label','Hx(f)', 'Tag','Plotuimenu4');
plotVin = uimenu('Parent',plotHndl, 'Separator','on', ...
    'Label','Vin', 'Tag','Plotuimenu5');
plotVint = uimenu('Parent',plotVin, 'Callback','nflldvis("plotVint")', ...
    'Label','Vin(t)', 'Tag','Plotuimenu6');
plotVinf = uimenu('Parent',plotVin, 'Callback','nflldvis("plotVinf")', ...
    'Label','|Vin(f)|', 'Tag','Plotuimenu7');
plotfildtrd = uimenu('Parent',plotHndl, 'Separator','on', ...
    'Label','"Filtered" Freq Domain Fields', 'Tag','Plotuimenu8');
plotEy2 = uimenu('Parent',plotfildtrd, 'Callback','nflldvis("plotEy2")', ...
    'Label','Ey(f)', 'Tag','Plotuimenu9');
plotEz2 = uimenu('Parent',plotfildtrd, 'Callback','nflldvis("plotEz2")', ...
    'Label','Ez(f)', 'Tag','Plotuimenu10');
plotHx2 = uimenu('Parent',plotfildtrd, 'Callback','nflldvis("plotHx2")', ...
    'Label','Hx(f)', 'Tag','Plotuimenu11');

%=====
% The Print Options menu & sub-menus
printHndl = uimenu('Label','Print', 'Enable','on', ...
    'Tag','PrintMenu');
prntscreen = uimenu('Parent',printHndl, 'Callback','nflldvis("save")', ...
    'Label','Save Figure as .TIF', 'Tag','Printuimenu1');
prntfig = uimenu('Parent',printHndl, 'Callback','nflldvis("printaxis1")', ...
    'Label','Print Figure Only', 'Tag','Printuimenu2');
prntfigncb = uimenu('Parent',printHndl, 'Callback','nflldvis("printallaxes")', ...
    'Label','Print Figure and ColorBar', 'Tag','Printuimenu3');

%=====
% The Colorbar Axes
cbaraxes=axes('Color',[0.6 0.6 0.6], 'Units','normalized', ...
    'Position',[0.795 0.3003663003663004 0.04 0.4926739926739927], ...

```

```

    'Visible','off', 'Tag','Axes2');
axes(axHndl) % Reset the current axes to axHndl for plotting/rendering

% Uncover the figure
hndlList=[infoHndl startHndl demoHndl movieHndl continuHndl resetHndl closeHndl, ...
    edittextHndl grppopupHndl frampopupHndl fpspopupHndl zscalslidHndl, ...
    zscaltextHndl1 zscaltextHndl2 zscaltextHndl3 azieditHndl azitextHndl, ...
    eleditHndl eltextHndl cscalslidHndl cscaltextHndl1 cscaltextHndl2, ...
    cscaltextHndl3 plotHndl plotEy1 plotEz1 plotHx1 plotVint plotVinf, ...
    plotEy2 plotEz2 plotHx2 printHndl cbaraxes prntscreen];
set(figNumber,'Visible','on', 'UserData',hndlList)
set(continuHndl,'Value',1)
watchoff(oldFigNumber);
figure(figNumber);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'info'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
helpwin(mfilename)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'start'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
infoHndl=hndlList(1); startHndl=hndlList(2); demoHndl=hndlList(3);
movieHndl=hndlList(4); continuHndl=hndlList(5); resetHndl=hndlList(6);
closeHndl=hndlList(7); edittextHndl=hndlList(8); grppopupHndl=hndlList(9);
plotHndl=hndlList(24); printHndl=hndlList(33); cbaraxes=hndlList(34);

cla
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')
set(resetHndl,'Enable','on')
set(plotHndl,'Enable','off')
text(0,0,'Select the .geo file containing the AWAS antenna geometry.', ...
    'HorizontalAlignment','center');
pause(1e-20)
set(edittextHndl,'String','Select the .geo file containing the AWAS antenna geometry...')
[geofile,filepath] = uigetfile('* .geo','AWAS Geometry Files');

```

```

% *** Reading AWAS Antenna Geometry Data ***

% WARNING: Ensure the following parameters are met in order to read the correct data
% from the .GEO file. The # of nodes below assumes that the angle between
% the y-axis and a line drawn between Nodes 1 & 3 accurately duplicates the
% angular dimension of the TEM horn. If this is not so, then the following
% section will require changes (as will the "filtering" code located
% immediately prior to the movie function).
geofid = fopen(geofile,'rt');
geohdr = fscanff(geofid,'%76c',1); % Skips the header data on the .GEO file
g1 = fscanff(geofid,'%3d',1); % Skips the first integer value
GndPlane = fscanff(geofid,'%3d',1); % Indicates presence of a Ground Plane
g2 = fscanff(geofid,'%3d',3); % Skips the next 3 integer values
node1 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #1
node2 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #2
node3 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #3

% *** Reading AWAS Near-Field Data ***

cla;
text(0,0,'Select the .nfl file containing the AWAS Near-Field data.', ...
'HorizontalAlignment','center');
pause(1e-20)
set(edittextHndl,'String','Select the .nfl file containing the AWAS Near-Field data...')
[filename,filepath] = uigetfile('*.nfl','AWAS Near-Field Files');
fid = fopen(filename,'rt');
Ng = fscanff(fid,'%3d',1); % Number of Position Groups
Fp = fscanff(fid,'%12e',2); % (f1, f2) Start/Stop Freq Pts
Ns = fscanff(fid,'%4d',1); % Number of AWAS Freq Steps
f1 = Fp(1)/1e6; f2 = Fp(2)/1e6; df = (f2-f1)/Ns; f = 0:df:f2;
cla; set(edittextHndl,'String','')
text(0,0.1,['Number of Groups: ',int2str(Ng)],'HorizontalAlignment','center')
text(0,0,['Number of AWAS frequency steps: ',int2str(Ns)], ...
'HorizontalAlignment','center')
text(0,-0.1,['f_1: ',num2str(f1),' MHz; df=: ',num2str(df), ...
' MHz; f_2: ',num2str(f2),' MHz'],'HorizontalAlignment','center')
axis([-1 1 -1 1])
AntennaType = fscanff(fid,'%3d',1); % The Antenna Type (Transmitter/Rcvr)
Nd = fscanff(fid,'%3d',2); % Skips the next 3 integer values.

if AntennaType==2, % For Rcvr Antennas Only!
E_theta_real = fscanff(fid,'%12e',1); % Real Incident E-Field fm Theta Direction
E_theta_imag = fscanff(fid,'%12e',1); % Imag Incident E-Field fm Theta Direction
E_phi_real = fscanff(fid,'%12e',1); % Real Incident E-Field fm Phi Direction
E_phi_imag = fscanff(fid,'%12e',1); % Imag Incident E-Field fm Phi Direction

```

```

theta = fscanf(fid,'%8f',2);      % (Theta1, Theta2) Start/Stop Theta Angles
Ntheta = fscanf(fid,'%4d',1);     % Number of Steps from Theta1 to Theta2
phi = fscanf(fid,'%8f',2);        % (Phi1, Phi2) Start/Stop Phi Angles
Nphi = fscanf(fid,'%4d',1);       % Number of Steps from Phi1 to Phi2
end

```

```

xp = zeros(Ng,2);  yp = zeros(Ng,2);  zp = zeros(Ng,2);

```

```

for ng=1:Ng;                % Reading Group Coordinates
    xp(ng,:) = fscanf(fid,'%12e',[1,2]); % (x1, x2) Near Fld Pts
    Nxs(ng) = fscanf(fid,'%4d',1);      % No. x-steps
    yp(ng,:) = fscanf(fid,'%12e',[1,2]); % (y1, y2) Near Fld Pts
    Nys(ng) = fscanf(fid,'%4d',1);      % No. y-steps
    zp(ng,:) = fscanf(fid,'%12e',[1,2]); % (z1, z2) Near Fld Pts
    Nzs(ng) = fscanf(fid,'%4d',1);      % No. z-steps
    grpstr(ng,:) = sprintf('%13.0f',ng); % Create a string array of nfld grps
end

```

```

popupstr1 = char('Near-Field Group',grpstr); % Creates the final near-field group list
set(grppopupHndl,'Enable','on','String',popupstr1)
set(continuhndl,'Enable','on')
set(edittextHndl,'String','Select the Near-Field group to analyze from the "Near-Field Group"
pop-up menu below. Press the "Continue" button once you have made your selection...')
waitfor(continuhndl,'Value')
set(edittextHndl,'String','Calculating the Frequency-Domain E- and H-Fields...')
watchon;
set(continuhndl,'Value',1,'Enable','off')
pause(1e-20)

```

```

NG = get(grppopupHndl,'Value');
set(grppopupHndl,'Enable','off')
if NG>1, NG=NG-1; end
x1 = xp(NG,1); x2 = xp(NG,2);
y1 = yp(NG,1); y2 = yp(NG,2);
z1 = zp(NG,1); z2 = zp(NG,2);
Nx = Nxs + ones(size(Nxs));  Ny = Nys + ones(size(Nys));
Nz = Nzs + ones(size(Nzs));  % Total # of x,y & z points
dx = (x2-x1)/Nxs(NG); x = x1:dx:x2;
dy = (y2-y1)/Nys(NG); y = y1:dy:y2;
dz = (z2-z1)/Nzs(NG); z = z1:dz:z2;

```

```

if z1==z2,                % AWAS near-field data is in the xy-plane
    N1 = Nx(NG); N1s = Nxs(NG); N2 = Ny(NG); N2s = Nys(NG);
    rowposit = 1:N2; colposit = (N1+1)/2; xdim = x; xlabl = 'x'; ydim = y;
    ylabl = 'y'; minxdim = x1; maxxdim = x2; minydim = y1; maxydim = y2;
else
    % AWAS near-field data is in the yz-plane

```



```

N1 = Ny(NG); N1s = Nys(NG); N2 = Nz(NG); N2s = Nzs(NG);
rowposit = (N2+1)/2; colposit = 1:N1; xdim = y; xlabel = 'y'; ydim = z;
ylabel = 'z'; minxdim = y1; maxxdim = y2; minydim = z1; maxydim = z2;
end
Nf = Ns+2; % Number of points to f2 folding freq (needs to be 2^m+1)
Nrow = 2*Nx.*Ny.*Nz; EH = zeros(6,Nrow(NG));
Eyf = zeros(N2,N1,Nf); if GndPlane==1 & z1~=z2, Eyf = zeros(2*N2s+1,N1,Nf); end
Ezf = Eyf; Hxf = Ezf;

for m=2:Nf; % Frequency Stepping
    for n=1:Ng; % Reading Group Data
        S = fgetl(fid); % Read blank line between EH data arrays
        EH = fscanf(fid,'%12e',[6,Nrow(n)]);
        if n == NG, % Saving Data at Selected Group
            Eyf(1:N2,.,m) = reshape(EH(3,1:2:Nrow(NG)-1)+j*EH(4,1:2:Nrow(NG)-1)',N2,N1);
            Ezf(1:N2,.,m) = reshape(EH(5,1:2:Nrow(NG)-1)+j*EH(6,1:2:Nrow(NG)-1)',N2,N1);
            Hxf(1:N2,.,m) = reshape(EH(1,2:2:Nrow(NG))+j*EH(2,2:2:Nrow(NG))',N2,N1);
        end
    end
end
end

if GndPlane==1
    Eyf=.5*Eyf; Ezf=.5*Ezf; Hxf=.5*Hxf; % Scale factor to comp for gndplane use
    if z1~=z2, % Changes the appropriate variables due to
        N2s=2*N2s; N2=N2s+1; z1=-z2; % the presence of the ground plane.
        dz=(z2-z1)/N2s; z=z1:dz:z2; rowposit=(N2+1)/2; ydim=z; minydim=z1;
        Eyf((N2+1)/2:N2,.,) = Eyf(1:(N2+1)/2,.,); % Reshape the Eyf matrix to
        Eyf(1:N2s/2,.,) = flipdim(Eyf((N2+3)/2:N2,.,),1); % fit the gnd plane "image"
        Ezf((N2+1)/2:N2,.,) = Ezf(1:(N2+1)/2,.,); % Reshape the Ezf matrix to
        Ezf(1:N2s/2,.,) = flipdim(Ezf((N2+3)/2:N2,.,),1); % fit the gnd plane "image"
        Hxf((N2+1)/2:N2,.,) = Hxf(1:(N2+1)/2,.,); % Reshape the Hxf matrix to
        Hxf(1:N2s/2,.,) = flipdim(Hxf((N2+3)/2:N2,.,),1); % fit the gnd plane "image"
    end
end
fclose(fid); fclose(geofid); clear EH

save Eyf Eyf
save Ezf Ezf
save Hxf Hxf
watchoff

% Retrieve the user's ID info for all plots:
set(edittextHndl,'String','Type brief ID information for all plots in the input box below...')
IDinfo = inputdlg({'Enter brief ID information for all plots:'}, ...
    'ID Info for All Plots',1);
IDinfo = char(IDinfo);

```

```

watchon;

% Create the Gaussian Impulse Input Voltage, Vin(t). & |Vin(f)|. the Fourier
% Transform of Vin(t):
set(edittextHndl,'String','Creating the Gaussian impulse input voltage...')
pause(1e-20)
Nt = 2*Nf-2;          % Number of radix-2 frequency and time points (2^m)
Tr = 1/df;   dt = Tr/Nt;   T = Tr-dt;   t = 0:dt:T;
Vint = zeros(1,Nt);   Vinf = Vint;
t1 = 0:29;
if AntennaType == 2
    sigma1 = 1;   sigma2 = 2;   A1 = sigma2/(sigma2-sigma1);   A2 = 1-A1;
    vin = A1*exp(-(t1-14).^2./(2*sigma1^2)) + A2*exp(-(t1-14).^2./(2*sigma2^2));
else
    vin = exp(-(t1-14).^2./2);
end
Vint(1:30) = vin;
fp = f(1:Nf);   Vinf = 1e-9*fft(Vint);   Vif = abs(Vinf(1:Nf));

% Frequency Scaling of AWAS Computed Fields Due to Gaussian Input Voltage
set(edittextHndl,'String','Scaling the E- and H-Fields due to the Gaussian Input Voltage...')
pause(1e-20)
Vif2 = zeros(1,1,Nt);          % Transform Vif from a vector to a
Vif2(1,1,:) = Vinf;            % 3-D array of N2 rows. 1 column. &
Vin = repmat(Vif2,[N2,1,1]);   % Nt "tiles" (the 3rd dimension)
Eyf=-Eyf;   Ezf=-Ezf;   Hxf=-Hxf; % Corrects for a 180 degree phase shift

if AntennaType == 2
    k = 2*pi/3e2*fp;           % Create a time delay for the case
    d = zeros(1,1,Nf);         % of the antenna acting as a receiver.
    yo = 3;                    % Initial location of incident wave.
    d(1,1,:) = exp(-j*yo*k);
    delay = repmat(d,[N2,1,1]);
    eta = 120*pi;              % Intrinsic Impedance (~377 ohms).
    Einc = zeros(N2,1,length(fp)); % Create the incident fields for the case
    Einc(:,1,:) = exp(j*y'*k); % of the antenna acting as a receiver.
    for m = 1:N1               % Calculate the total fields.
        Eyf(:,m,:) = Vin(:,1,1:Nf).*delay.*Eyf(:,m,:);
        Ezf(:,m,:) = Vin(:,1,1:Nf).*delay.*Ezf(:,m,:);
        Hxf(:,m,:) = Vin(:,1,1:Nf).*delay.*Hxf(:,m,:);
    end
else
    for m = 1:N1
        Eyf(:,m,:) = Vin(:,1,1:Nf).*Eyf(:,m,:);
        Ezf(:,m,:) = Vin(:,1,1:Nf).*Ezf(:,m,:);
        Hxf(:,m,:) = Vin(:,1,1:Nf).*Hxf(:,m,:);
    end
end

```

```

end
end

save Hdr f y rowposit colposit Nf IDinfo T t Vint fp Vif
save Eyff Eyf          % Saving variables for all "Plot" functions
save Ezff Ezf
save Hxff Hxf

% Convert the Frequency-Domain Data to a Time-Domain Representation
set(edittextHndl,'String','Calculating the Time-Domain Fields...')
pause(1e-20)
A1 = zeros(N2,N1,Nt);   A2 = A1;   A3 = A1;
Eyt = A1;  Ezt = A1;  Hxt = A1;  fsc1 = 2e6*f2;
A1(:, :, 1:Nf) = Eyf;  A2(:, :, 1:Nf) = Ezf;  A3(:, :, 1:Nf) = Hxf;
clear Einc Eyf Ezf Hxf Vif Vint Vinf Vin k d delay % Free up some memory
A1(:, :, Nf:Nt) = A1(:, :, Nf:Nt) + conj(flipdim(A1(:, :, 2:Nf), 3));
A2(:, :, Nf:Nt) = A2(:, :, Nf:Nt) + conj(flipdim(A2(:, :, 2:Nf), 3));
A3(:, :, Nf:Nt) = A3(:, :, Nf:Nt) + conj(flipdim(A3(:, :, 2:Nf), 3));

for m=1:N2
    for n=1:N1
        Eyt(m,n,:) = fsc1*real(iff(A1(m,n,:)));
        Ezt(m,n,:) = fsc1*real(iff(A2(m,n,:)));
        Hxt(m,n,:) = fsc1*real(iff(A3(m,n,:)));
    end
end
clear A1 A2 A3          % Free up more memory

% "Filter" the Eyt, Ezt, & Hxt Data for the YZ-Plane Case. This
% is done to remove the high values of the fields directly on top
% of the wire and those outside the wire structure:
if x1 == x2,
    a = (node3(3)-node1(3))/(node3(2)-node1(2)); % a=dz/dy, slope of "V"
    b = -a*y1; % Offset of line to ensure passage thru 1st y point of near-field
    Z1 = a*y + b*ones(size(y));
    Z2 = -Z1;
    M = zeros(size(Eyt));  M1 = zeros(N2,N1);
    for N = 1:Ny;
        M1(:,N) = flipr(z)'>Z1(N) | flipr(z)'<Z2(N);
    end
    M = repmat(M1,[1 1 Nt]);
    Eyt = Eyt.*~M;  Ezt = Ezt.*~M;  Hxt = Hxt.*~M;
end; clear M M1 M2
watchoff

save Movihdr T Nt dt minxdim maxxdim minydim maxydim xdim ydim xlabl ylabl

```

```

save Eyt Eyt          % Saving variables for use with "Movie" function
save Ezt Ezt
save Hxt Hxt

clear Eyt Ezt Hxt      % Free up more memory
set(edittextHndl,'String','')
set(grppopupHndl,'Value',1)
set(figNumber,'UserData',hndlList);
cla;
text(0,0.1,'All calculations are complete.','HorizontalAlignment','center')
text(0,0,'Press the "Movie" button to create a movie of the time-domain fields,', ...
    'HorizontalAlignment','center')
text(0,-0.1,'or plot the frequency-domain representation of the fields using the "Plot" pull-down
menu at the top of this window.', ...
    'HorizontalAlignment','center')
axis([-1 1 -1 1])
set(movieHndl,'Enable','on')
set(plotHndl,'Enable','on')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'demo'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber =(gcf);
hndlList = get(figNumber,'UserData');
infoHndl=hndlList(1); startHndl=hndlList(2); demoHndl=hndlList(3);
movieHndl=hndlList(4); continuHndl=hndlList(5); resetHndl=hndlList(6);
closeHndl=hndlList(7); edittextHndl=hndlList(8); grppopupHndl=hndlList(9);
frampopupHndl=hndlList(10); fspopupHndl=hndlList(11); zscalslidHndl=hndlList(12);
azieditHndl=hndlList(16); eleditHndl=hndlList(18); cscalslidHndl=hndlList(20);
plotHndl=hndlList(24); printHndl=hndlList(33); cbaraxes=hndlList(34);

cla
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')
set(resetHndl,'Enable','on')
set(grppopupHndl,'Enable','on')
set(plotHndl,'Enable','off')
text(0,0,'Select the .geo file containing the AWAS antenna geometry.', ...
    'HorizontalAlignment','center');
pause(1e-20)
set(edittextHndl,'String','Select the .geo file containing the AWAS antenna geometry...')
[geofile,filepath] = uigetfile('demo.geo','AWAS Geometry Files');

```

```

if ~strcmp(geofile,'Demo.geo')
    er = errordlg('You may only select the "Demo.geo" file for this demo!','ERROR');
    waitfor(er,'ButtonDownFcn')
end
geofile = 'Demo.geo';

%          *** Reading AWAS Antenna Geometry Data ***

% WARNING: Ensure the following parameters are met in order to read the correct data
%   from the .GEO file. The # of nodes below assumes that the angle between
%   the y-axis and a line drawn between Nodes 1 & 3 accurately duplicates the
%   angular dimension of the TEM horn. If this is not so, then the following
%   section will require changes (as will the "filtering" code located
%   immediately prior to the movie function).
geofid = fopen(geofile,'rt');
geohdr = fscanff(geofid,'%76c',1); % Skips the header data on the .GEO file
g1 = fscanff(geofid,'%3d',1); % Skips the first integer value
GndPlane = fscanff(geofid,'%3d',1); % Indicates presence of a Ground Plane
g2 = fscanff(geofid,'%3d',3); % Skips the next 3 integer values
node1 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #1
node2 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #2
node3 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #3

%          *** Simulated Reading of AWAS Near-Field Data ***

cla;
text(0,0,'Select the .nfl file containing the AWAS Near-Field data.', ...
    'HorizontalAlignment','center');
pause(1e-20)
set(edittextHndl,'String','Select the .nfl file containing the AWAS Near-Field data...')
[filename,filepath] = uigetfile('demo.nfl','AWAS Near-Field Files');
cla;
text(0,0.1,'Number of Groups: 1','HorizontalAlignment','center')
text(0,0,'Number of AWAS frequency steps: 127', ...
    'HorizontalAlignment','center')
text(0,-0.1,'f_1: 4 MHz; df=: 4 MHz; f_2: 512 MHz', ...
    'HorizontalAlignment','center')
axis([-1 1 -1 1])

grpstr = sprintf('%13.0f',1); % Create a string array of nflid grps
popupstr1 = char('Near-Field Group',grpstr);
set(grppopupHndl,'String',popupstr1) % Creates the final near-field group list
set(continuhndl,'Enable','on')
set(edittextHndl,'String','Select the Near-Field group to analyze from the "Near-Field Group"
pop-up menu below. Press the "Continue" button once you have made your selection...')
waitfor(continuhndl,'Value')

```

```

set(edittextHndl,'String','Calculating the Frequency-Domain E- and H-Fields...')
watchon;
set(continuhndl,'Value',1,'Enable','off')
set(grppopuhndl,'Enable','off')
pause(5)
fclose(geofid);
load DemoHdr, load DemoEzf
watchoff

set(edittextHndl,'String','Type brief ID information for all plots in the input box below...')
IDinfo = inputdlg({'Enter brief ID information for all plots:'}, ...
    'ID Info for All Plots',1);
IDinfo = char(IDinfo);
pause(1e-20)

% Plot a sample of the "Impulse Response" Fields in the frequency domain:
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Ezf(rowposit,colposit,1:Nf))))
title(['Near-Field |E_z(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_z| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Ezf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
Data=1:64;Data=(Data*Data)/64;
h=msgbox([' Place the mousepointer on the plot, then press ';'and hold the left mouse button
to rotate the plot.'],'To Rotate Plot:','custom',Data,hot(64));
set(continuhndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of the E-Field in the frequency-domain.','Press the
"Continue" button once you are through with this plot...'))
pause(1e-20)
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
rotate3d off
set(cbaraxes,'Visible','off')
set(axHndl,'Visible','off')
axes(axHndl)

% Plot the Gaussian Impulse Input Voltage, Vin(t):
cla reset; plot(t,Vint), title(['Antenna Input Voltage V_{in}(t): ',IDinfo]);
xlabel('Time (uSec)'), ylabel('Volts'), axis([0 T -0.1 max(Vint)+0.1])
set(continuhndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of the Gaussian impulse input voltage,
Vin(t).','Press the "Continue" button once you are through with this plot...'))
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')

```

```

% Plot |Vin(f)|, the Fourier Transform of Vin(t):
cla; plot(fp,Vif), title(['Antenna Input Voltage |V_{in}(f)|: ',IDinfo])
xlabel('Freq (MHz)'), axis([0 f(Nf) 0 max(Vif)])
set(continHndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of |Vin(f)|, the frequency-domain representation of
Vin(t).','Press the "Continue" button once you are through with this plot...'))
waitfor(continHndl,'Value')
set(continHndl,'Value',1,'Enable','off')
set(edittextHndl,'String','Scaling the E- and H-Fields due to the Gaussian Input Voltage...')
watchon;
pause(5)
watchoff

```

```

% Displaying a sample of the "Filtered" Frequency-Domain Fields vs y:
load DemoEzff
cla; set(axHndl,'Visible','on'), set(cbaraxes,'Visible','on')
mesh(f,y,squeeze(abs(Ezf(rowposit,colposit,1:Nf))))
title(['Near-Field |E_z(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_z| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Ezf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
h=msgbox([' Place the mousepointer on the plot, then press ','and hold the left mouse button
to rotate the plot.'],'To Rotate Plot:','custom',Data,hot(64));
set(continHndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of the E-Field due to the Gaussian input
voltage.','Press the "Continue" button once you are through with this plot...'))
pause(1e-20)
waitfor(continHndl,'Value');
set(continHndl,'Value',1,'Enable','off')
rotate3d off
set(axHndl,'Visible','off')
axes(axHndl)
set(edittextHndl,'String','Calculating the Time-Domain Fields...')
watchon;
pause(5)
watchoff

```

```

% Generate & Display the Near-Fields Ey(t), Ez(t), & Hx(t) Using
% the Movie Format:
clear Ezf
load DemoMovihdr, load DemoEzt
Tns = 1000*T; dt = Tns/(Nt-1); t = 0:dt:Tns; % Time in nanoseconds
frames = 0:50:Nt;
for n=1:length(frames)-1
    framestr(n,:) = sprintf('%13.0f',frames(n+1));

```

```

end
popupstr3 = char('Total Frames',framestr);
set(frampopupHndl,'String',popupstr3) % Creates the Total Frames pop-up list
cla;
text(0,0.1,['T = ',num2str(T),' uSec'])
text(0,0,['N = ',int2str(Nt),' Samples'])
text(0,-0.1,['dt = ',num2str(dt),' nSec'])
axis([-1 1 -1 1])
set(frampopupHndl,'Enable','on','String',popupstr3)
set(continuhndl,'Enable','on')
set(edittextHndl,'String','Select the number of frames for the movie from the "Total Frames"
pop-up menu below. Press the "Continue" button once you have made your selection...')
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
Mt = 50*(get(frampopupHndl,'Value')-1);
if Mt==0, Mt=25; end
set(fpspopupHndl,'Enable','on')
set(continuhndl,'Enable','on')
set(edittextHndl,'String','Select the frame rate of the movie from the "Frames-per-sec" pop-up
menu below. Press the "Continue" button once you have made your selection...')
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
fps = get(fpspopupHndl,'Value');
if fps>=3, fps=(fps-1)*2; end
set(frampopupHndl,'Enable','off')
set(fpspopupHndl,'Enable','off')
set(edittextHndl,'String','Now, create a movie to illustrate the E- or H-Field in the time domain.
Use this frame from your movie to create the axes, coloraxis, and view you desire...')
pause(5)
cla; set(axHndl,'Visible','on')
rotate3d on % Creates a moveable mesh plot for scaling axes prior to movie creation
Fieldmax = max(max(max(Ezt))); Fieldmin = min(min(min(Ezt)));
mesh(xdim,ydim,Ezt(:,:,25))
vmax = max(abs(Fieldmax),abs(Fieldmin));
vaxis = [minxdim maxxdim minydim maxydim -vmax vmax];
axis(vaxis), cmatrix = [flipud(jet(64)); jet(64)];
colormap(cmatrix), caxis([-vmax vmax])
set(cbaraxes,'YLim',[-vmax vmax]), colorbar(cbaraxes), axes(axHndl)
set(zscalslidHndl,'Enable','on','Value',1)
set(azieditHndl,'Enable','on')
set(eleditHndl,'Enable','on')
set(cscalslidHndl,'Enable','on','Value',1)
while 1 % Sets up the loop to customize display of data
    set(continuhndl,'Enable','on')
    set(edittextHndl,'String','To customize your movie, adjust the z-axis & c-axis sliders at the
bottom of the window. Press the "Continue" button once you have made your adjustments...')

```



```

waitfor(continuhndl,'Value')
set(continuhndl,'Enable','off','Value',1)
set(edittextHndl,'String','')
zscale = get(zscalslidHndl,'Value'); if zscale==0, zscale=0.01; end
cscale = get(cscalslidHndl,'Value'); if cscale==0, cscale=0.01; end
[azi el] = view; % Reads the adjusted azi & el
set(azieditHndl,'String',num2str(azi))
set(eleditHndl,'String',num2str(el))
vaxis = [minxdim maxxdim minydim maxydim -zscale*vmax zscale*vmax];
mesh(xdim,ydim,Ezt(:, :, 25))
view([azi el]), axis(vaxis)
cmatrix = [flipud(jet(64)); jet(64)]; colormap(cmatrix)
axHndl=gca; caxis([-cscale*vmax cscale*vmax])
set(cbaraxes,'YLim',[-cscale*vmax cscale*vmax]), colorbar(cbaraxes); axes(axHndl)
mulligan = questdlg('Would you like to make further adjustments to the plot?', 'Continue
Adjustments?', 'No');
    if strcmp(mulligan,'No'), break, end
end
rotate3d off
set(zscalslidHndl,'Enable','off')
set(azieditHndl,'Enable','off')
set(eleditHndl,'Enable','off')
set(cscalslidHndl,'Enable','off')

cla;
FieldMovi = moviein(Mt); % Create the movie
set(gca,'NextPlot','replacechildren')
for nt = 1:Mt
    cla;
    mesh(xdim,ydim,Ezt(:, :, nt))
    view([azi el]), axis(vaxis)
    caxis([-cscale*vmax cscale*vmax])
    xlabel([xlabel, ' (meters)']), ylabel([ylabel, ' (meters)'])
    zlabel('Ez(t) (V/m)'), title(['Frame Time: ', num2str(t(nt)), ' nSec'])
    FieldMovi(:, nt) = getframe;
end
title(' '),
movie(FieldMovi, 1, fps);

while 1
    mulligan = questdlg('Would you like to view the movie again?', ...
        'Instant Replay', 'No');
    if strcmp(mulligan,'No'), break, end
    movie(FieldMovi, 1, fps);
end

```

```

set(grppopupHndl,'Value',1)
set(fpspopupHndl,'Value',1)
set(frampopupHndl,'Value',1)
set(figNumber,'UserData',hndlList);
cla; set(axHndl,'Visible','off')
text(0,0,'That concludes the demo...', ...
    'HorizontalAlignment','center')
axis([-1 1 -1 1])
pause(2)
clear Ezt FieldMovi
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'movie'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cla;
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
infoHndl=hndlList(1); startHndl=hndlList(2); demoHndl=hndlList(3);
movieHndl=hndlList(4); continuHndl=hndlList(5); resetHndl=hndlList(6);
closeHndl=hndlList(7); edittextHndl=hndlList(8); grppopupHndl=hndlList(9);
frampopupHndl=hndlList(10); fpspopupHndl=hndlList(11); zscalslidHndl=hndlList(12);
azieditHndl=hndlList(16); eleditHndl=hndlList(18); cscalslidHndl=hndlList(20);
plotHndl=hndlList(24); printHndl=hndlList(33); cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')
set(plotHndl,'Enable','off')
movieMode = questdlg('Do you want to create a new movie or replay a saved one?', ...
    'What kind of movie?','Create Movie','Replay','Replay');

if strcmp(movieMode,'Replay')
    cla;
    text(0,0,'Select the .mat file containing the recorded movie.', ...
        'HorizontalAlignment','center')
    pause(1e-20)
    set(edittextHndl,'String','Select the .mat file containing the recorded movie...')
    [filename,filepath] = uigetfile('*.mat','Movie Files');
    cla; set(edittextHndl,'String',['Loading the movie file, ',filename]), pause(1e-20)
    eval(['load ',filepath,filename,' -mat']);
    set(fpspopupHndl,'Enable','on')
    set(continuHndl,'Enable','on')

```

```

    set(edittextHndl,'String','Select the frame rate of the movie from the "Frames-per-sec" pop-up
menu below. Press the "Continue" button once you have made your selection...')
    waitfor(continuhndl,'Value')
    set(continuhndl,'Value',1,'Enable','off')
    fps = get(fpspopupHndl,'Value');
    if fps>=3, fps=(fps-1)*2; end
    set(fpspopupHndl,'Enable','off')
    set(azieditHndl,'String',num2str(azi))
    set(eleditHndl,'String',num2str(el))
    axis(vaxis), axis('on'), view([azi el])
    xlabel([xlabl,' (meters)']), ylabel([ylabl,' (meters)']), zlabel(zlabl);
    colormap(cmatrix)
    caxis([-cscale*vmax cscale*vmax]), set(cbaraxes,'YLim',[-cscale*vmax cscale*vmax])
    colorbar(cbaraxes), axes(axHndl)
    wnumber = watchon;
    movie(axHndl,FieldMovi,1,fps);
    watchoff(wnumber);
    while 1
        mulligan = questdlg('Would you like to view the movie again?', ...
            'Instant Replay','No');
        if strcmp(mulligan,'No'), break, end
        movie(FieldMovi,1,fps);
    end
    cla; set(axHndl,'Visible','off');
    set(fpspopupHndl,'Value',1);
    set(figNumber,'Visible','on', 'UserData',hndlList);
    clear FieldMovi
    eval('nflavis("reset")');

else
    text(0,0,'Select the .mat file containing the header data (default is MOVIHDR.MAT)...',
'HorizontalAlignment','center')
    axis([-1 1 -1 1])
    pause(1e-20)
    set(edittextHndl,'String','Select the .mat file containing the header data (default is
MOVIHDR.MAT)...')
    [filename,filepath] = uigetfile('movihdr.mat','Movie Header Data File');
    eval(['load ',filepath,filename,' -mat']);
    set(edittextHndl,'String','')
    cla;
    filetype = questdlg('Select the E- or H-Field you want to view:','Field
Type','Eyt','Ezt','Hxt','Ezt');
    text(0,0,'Select the .mat file containing the time-domain E- or H-Fields (EYT.MAT,
EZT.MAT, or HXT.MAT)...', 'HorizontalAlignment','center');
    pause(1e-20)

```

```

    set(edittextHndl,'String','Select the .mat file containing the time-domain E- or H-Fields
(EYT.MAT, EZT.MAT, or HXT.MAT)...')
    [filename,filepath] = uigetfile([filetype '.mat'],'Ey(t), Ez(t), or Hx(t) Data File');
    eval(['load ',filepath,filename,' -mat']);
    set(edittextHndl,'String','')

% Generate & Display the Near-Fields Ey(t), Ez(t), & Hx(t) Using the Movie Format:
    Tns = 1000*T;    dt = Tns/(Nt-1);    t = 0:dt:Tns;    % Time in nanoseconds
    frames = 0:50:Nt;
    for n=1:length(frames)-1
        framestr(n,:) = sprintf('%13.0f',frames(n+1));
    end
    popupstr3 = char('Total Frames',framestr);
    set(frampopupHndl,'String',popupstr3)    % Creates the Total Frames pop-up list

while 1                                % Loop for adjusting the parameters of the movie
    cla;
    text(0,0.1,['T = ',num2str(T),' uSec'])
    text(0,0,['N = ',int2str(Nt),' Samples'])
    text(0,-0.1,['dt = ',num2str(dt),' nSec'])
    axis([-1 1 -1 1])
    set(frampopupHndl,'Enable','on','String',popupstr3)
    set(continuhndl,'Enable','on')
    set(edittextHndl,'String','Select the number of frames for the movie from the "Total Frames"
pop-up menu below. Press the "Continue" button once you have made your selection...')
    waitfor(continuhndl,'Value')
    set(continuhndl,'Value',1,'Enable','off')
    Mt = 50*(get(frampopupHndl,'Value')-1);    % Total # of frames to use in Movie
    if Mt==0, Mt=25; end
    set(fpspopupHndl,'Enable','on')
    set(continuhndl,'Enable','on')
    set(edittextHndl,'String','Select the frame rate of the movie from the "Frames-per-sec" pop-
up menu below. Press the "Continue" button once you have made your selection...')
    waitfor(continuhndl,'Value')
    set(continuhndl,'Value',1,'Enable','off')
    fps = get(fpspopupHndl,'Value');    % Frames-per-sec for the Movie
    if fps>=3, fps=(fps-1)*2; end
    set(frampopupHndl,'Enable','off')
    set(fpspopupHndl,'Enable','off')
    set(edittextHndl,'String','This is a sample mesh plot from your movie with the default
settings...')
    pause(2)
    cla; set(axHndl,'Visible','on')
    rotate3d on % Creates a moveable mesh plot for scaling axes prior to movie creation
    if strcmp(filetype,'Eyt')
        Fieldmax = max(max(max(Eyt)));    Fieldmin = min(min(min(Eyt)));

```

```

    mesh(xdim,ydim,Eyt(:, :, 37)),    zlabl = 'E_y(t) (V/m)';
    xlabel([xlabl, ' (meters)']), ylabel([ylabl, ' (meters)'])
    ylabel([zlabl]), title(['Frame Time: ', num2str(t(37)), ' nSec'])
elseif strcmp filetype, 'Ezt')
    Fieldmax = max(max(max(Ezt)));    Fieldmin = min(min(min(Ezt)));
    mesh(xdim,ydim,Ezt(:, :, 37)),    zlabl = 'E_z(t) (V/m)';
    xlabel([xlabl, ' (meters)']), ylabel([ylabl, ' (meters)'])
    ylabel([zlabl]), title(['Frame Time: ', num2str(t(37)), ' nSec'])
else
    Fieldmax = max(max(max(Hxt)));    Fieldmin = min(min(min(Hxt)));
    mesh(xdim,ydim,Hxt(:, :, 37)),    zlabl = 'H_x(t) (A/m)';
    xlabel([xlabl, ' (meters)']), ylabel([ylabl, ' (meters)'])
    ylabel([zlabl]), title(['Frame Time: ', num2str(t(37)), ' nSec'])
end
azi = str2num(get(azieditHndl, 'String'));
el = str2num(get(eleditHndl, 'String'));
view([azi el]) % If rebuilding the movie, this holds the last view parameters
vmax = max(abs(Fieldmax), abs(Fieldmin));
vaxis = [minxdim maxxdim minydim maxydim -vmax vmax];
axis(vaxis), cmatrix = [flipud(jet(64)); jet(64)];
colormap(cmatrix), caxis([-vmax vmax])
set(cbaraxes, 'YLim', [-vmax vmax]), colorbar(cbaraxes), axes(axHndl)
set(zcalslidHndl, 'Enable', 'on', 'Value', 1)
set(azieditHndl, 'Enable', 'on')
set(eleditHndl, 'Enable', 'on')
set(cscalslidHndl, 'Enable', 'on', 'Value', 1)
while 1 % Sets up the loop to customize display of data
    set(continuhndl, 'Enable', 'on')
    set(edittextHndl, 'String', 'To customize your movie, adjust the z-axis & c-axis sliders at the
bottom of the window. Press the "Continue" button once you have made your adjustments...')
    waitfor(continuhndl, 'Value')
    set(continuhndl, 'Enable', 'off', 'Value', 1)
    set(edittextHndl, 'String', '')
    zscale = get(zcalslidHndl, 'Value'); if zscale==0, zscale=0.01; end
    cscale = get(cscalslidHndl, 'Value'); if cscale==0, cscale=0.01; end
    [azi el] = view; % Reads the adjusted azi & el
    set(azieditHndl, 'String', num2str(azi))
    set(eleditHndl, 'String', num2str(el))
    vaxis = [minxdim maxxdim minydim maxydim -zscale*vmax zscale*vmax];
    if strcmp filetype, 'Eyt')
        mesh(xdim,ydim,Eyt(:, :, 37))
        xlabel([xlabl, ' (meters)']), ylabel([ylabl, ' (meters)'])
        ylabel([zlabl]), title(['Frame Time: ', num2str(t(37)), ' nSec'])
    elseif strcmp filetype, 'Ezt')
        mesh(xdim,ydim,Ezt(:, :, 37))
        xlabel([xlabl, ' (meters)']), ylabel([ylabl, ' (meters)'])

```

```

        xlabel([xlabel]), title(['Frame Time: ',num2str(t(37)),' nSec'])
    else
        mesh(xdim,ydim,Hxt(:,37))
        xlabel([xlabel,' (meters)']), ylabel([ylabel,' (meters)'])
        xlabel([xlabel]), title(['Frame Time: ',num2str(t(37)),' nSec'])
    end
    view([azi el]), axis(vaxis)
    cmatrix = [flipud(jet(64)); jet(64)]; colormap(cmatrix)
    axHndl=gca; caxis([-cscale*vmax cscale*vmax])
    set(cbaraxes,'YLim',[-cscale*vmax cscale*vmax]), colorbar(cbaraxes); axes(axHndl)
    mulligan = questdlg('Would you like to make further adjustments to the plot?', 'Continue
Adjustments?', 'No');
    if strcmp(mulligan,'No'), break, end
end
rotate3d off
set(zscalslidHndl,'Enable','off')
set(azieditHndl,'Enable','off')
set(eleditHndl,'Enable','off')
set(cscalslidHndl,'Enable','off')

cla; % Create the movie
FieldMovi = moviein(Mt);
set(gca,'NextPlot','replacechildren')
for nt = 1:Mt
    cla;
    if strcmp filetype,'Eyt')
        mesh(xdim,ydim,Eyt(:,nt))
    elseif strcmp filetype,'Ezt')
        mesh(xdim,ydim,Ezt(:,nt))
    else
        mesh(xdim,ydim,Hxt(:,nt))
    end
    view([azi el]), axis(vaxis)
    caxis([-cscale*vmax cscale*vmax])
    xlabel([xlabel,' (meters)']), ylabel([ylabel,' (meters)'])
    xlabel([xlabel]), title(['Frame Time: ',num2str(t(nt)),' nSec'])
    FieldMovi(:,nt) = getframe;
end
title(' '),
movie(FieldMovi,1,fps);
while 1
    mulligan = questdlg('Would you like to view the movie again?', ...
        'Instant Replay','No');
    if strcmp(mulligan,'No'), break, end
    movie(FieldMovi,1,fps);
end

```

```

        rebuild = questdlg('Would you like to adjust any of the movie parameters and build it
again?','Rebuild?','No');
        cla; set(axHndl,'Visible','off');
        set(fpspopupHndl,'Value',1)
        set(frampopupHndl,'Value',1)
        if strcmp(rebuild,'No'), break, end
    end
    % Now save the movie for replay later
    [newmoviename, newmoviepath] = uiputfile('*.mat', 'Save Movie As');

save(newmoviename,'FieldMovi','vaxis','xlabl','ylabl','zlabl','cmatrix','zscale','cscale','vmax','azi',
el')
    clear Eyt Ezt Hxt FieldMovi
    eval('nflldvis("reset")');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotEyl'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    axHndl = gca;
    figNumber =(gcf);
    hndlList = get(figNumber,'UserData');
    startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
    continuHndl=hndlList(5); edittextHndl=hndlList(8); plotEyl=hndlList(25);
    cbaraxes=hndlList(34);
    set(startHndl,'Enable','off')
    set(demoHndl,'Enable','off')
    set(movieHndl,'Enable','off')

% Plot the "Impulse Response" y-component of the E-Field in the frequency domain:
load Eyf, load Hdr
set(edittextHndl,'String',char('This is the frequency-domain representation of the E-
Field.','Press the "Continue" button once you are through with this plot...'))
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Eyf(rowposit,colposit,1:Nf))))
title(['Near-Field |E_y(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_y| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Eyf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continuHndl,'Enable','on')
waitfor(continuHndl,'Value')
set(continuHndl,'Value',1,'Enable','off')
rotate3d off
set(axHndl,'Visible','off')

```

```

axes(axHndl)
clear Eyf
eval('nflvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotEz1'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuhndl=hndlList(5); edittextHndl=hndlList(8); plotEz1=hndlList(26);
cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the "Impulse Response" z-component of the E-Field in the frequency domain:
load Ezf, load Hdr
set(edittextHndl,'String',char("This is the frequency-domain representation of the E-
Field. 'Press the "Continue" button once you are through with this plot...'))
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Ezf(rowposit,colposit,1:Nf))))
title(['Near-Field |E_z(y,f)| for x=0 m; z=0 m; 'IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_z| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Ezf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continuhndl,'Enable','on')
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
rotate3d off
set(axHndl,'Visible','off')
axes(axHndl)
clear Ezf
eval('nflvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotHx1'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;

```



```

hdlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuHndl=hndlList(5); edittextHndl=hndlList(8); plotHx1=hndlList(27);
cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the "Impulse Response" x-component of the H-Field in the frequency domain:
load Hxf, load Hdr
set(edittextHndl,'String',char('This is the frequency-domain representation of the H-
Field.','Press the "Continue" button once you are through with this plot...'))
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Hxf(rowposit,colposit,1:Nf))))
title(['Near-Field |H_x(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|H_x| (A/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Hxf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continuHndl,'Enable','on')
waitfor(continuHndl,'Value')
set(continuHndl,'Value',1,'Enable','off')
rotate3d off
set(axHndl,'Visible','off')
axes(axHndl)
clear Hxf
eval('nflvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotVint'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hdlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuHndl=hndlList(5); edittextHndl=hndlList(8); plotVint=hndlList(28);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the Gaussian Impulse Input Voltage, Vin(t):
load Hdr
cla reset; set(axHndl,'Visible','on')
plot(t,Vint), title(['Antenna Input Voltage V_{in}(t): ',IDinfo]);

```

```

xlabel('Time (uSec)'), ylabel('Volts'), axis([0 T min(Vint)-0.1 max(Vint)+0.1])
set(continuhndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of the Gaussian impulse input voltage,
Vin(t).','Press the "Continue" button once you are through with this plot...'))
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
set(axHndl,'Visible','off')
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotVinf'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuhndl=hndlList(5); edittextHndl=hndlList(8); plotVinf=hndlList(29);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot |Vin(f)|, the Fourier Transform of Vin(t):
load Hdr
cla reset; set(axHndl,'Visible','on')
plot(fp,Vif), title(['Antenna Input Voltage |V_{in}(f)|: ',IDinfo])
xlabel('Freq (MHz)'), axis([0 f(Nf) 0 max(Vif)+0.1*max(Vif)])
set(continuhndl,'Enable','on')
set(edittextHndl,'String',char('This is a plot of |Vin(f)|, the frequency-domain representation of
Vin(t).','Press the "Continue" button once you are through with this plot...'))
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
set(axHndl,'Visible','off')
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotEy2'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuhndl=hndlList(5); edittextHndl=hndlList(8); plotEy2=hndlList(30);

```

```

cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the "Filtered" y-component of the E-Field in the frequency domain:
load Eyff, load Hdr
set(edittextHndl,'String','The frequency-domain representation of the E-Field due to the
Gaussian input voltage. Press the "Continue" button once you are through with this plot...')
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Eyff(rowposit,colposit,1:Nf))))
title(['Near-Field |E_y(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_y| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Eyff(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continHndl,'Enable','on')
waitfor(continHndl,'Value')
set(continHndl,'Value',1,'Enable','off')
rotate3d off
set(axHndl,'Visible','off')
axes(axHndl)
clear Eyf
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotEz2'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continHndl=hndlList(5); edittextHndl=hndlList(8); plotEz2=hndlList(35);
cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the "Filtered" z-component of the E-Field in the frequency domain:
load Ezff, load Hdr
set(edittextHndl,'String','The frequency-domain representation of the E-Field due to the
Gaussian input voltage. Press the "Continue" button once you are through with this plot...')
cla; set(axHndl,'Visible','on')
mesh(f,y,squeeze(abs(Ezff(rowposit,colposit,1:Nf))))

```

```

title(['Near-Field |E_z(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|E_z| (V/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Ezf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continuhndl,'Enable','on')
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
rotate3d off
set(axhndl,'Visible','off')
axes(axhndl)
clear Ezf
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'plotHx2'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axhndl = gca;
figNumber =(gcf);
hdlList = get(figNumber,'UserData');
startHndl=hndlList(2); demoHndl=hndlList(3); movieHndl=hndlList(4);
continuhndl=hndlList(5); edittextHndl=hndlList(8); plotHx2=hndlList(32);
cbaraxes=hndlList(34);
set(startHndl,'Enable','off')
set(demoHndl,'Enable','off')
set(movieHndl,'Enable','off')

% Plot the "Filtered" x-component of the H-Field in the frequency domain:
load Hxff, load Hdr
set(edittextHndl,'String','The frequency-domain representation of the H-Field due to the
Gaussian input voltage. Press the "Continue" button once you are through with this plot...')
cla; set(axhndl,'Visible','on')
mesh(f,y,squeeze(abs(Hxf(rowposit,colposit,1:Nf))))
title(['Near-Field |H_x(y,f)| for x=0 m; z=0 m; ',IDinfo])
ylabel('y (meters)'), xlabel('Freq (MHz)'), zlabel('|H_x| (A/(m-Hz))')
axis([0 max(f) 0 max(y) 0 1.1*max(max(squeeze(abs(Hxf(rowposit,colposit,1:Nf))))))]
rotate3d on
colorbar(cbaraxes);
set(continuhndl,'Enable','on')
waitfor(continuhndl,'Value')
set(continuhndl,'Value',1,'Enable','off')
rotate3d off
set(axhndl,'Visible','off')
axes(axhndl)

```

```

clear Hxf
eval('nflldvis("reset")');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'save'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[X,map]=capture(1);
plotname = inputdlg({'Enter the filename under which you want to save this figure:'}, ...
    '.TIF File Name',1);
plotname = char(plotname);
imwrite(X,map,[plotname,'.tif'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'printaxis1'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Data=1:64;Data=(Data*Data)/64;
h=msgbox(['Sorry, this function is not supported by MATLAB 5.1.'],'Reserved for Future
Release','custom',Data,hot(64));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'printallaxes'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Data=1:64;Data=(Data*Data)/64;
h=msgbox(['Sorry, this function is not supported by MATLAB 5.1.'],'Reserved for Future
Release','custom',Data,hot(64));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 'reset'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axHndl = gca;
figNumber = gcf;
hndlList = get(figNumber,'UserData');
infoHndl=hndlList(1); startHndl=hndlList(2); demoHndl=hndlList(3);
movieHndl=hndlList(4); continuHndl=hndlList(5); resetHndl=hndlList(6);
closeHndl=hndlList(7); edittextHndl=hndlList(8); grppopupHndl=hndlList(9);
frampopupHndl=hndlList(10); fpspopupHndl=hndlList(11); zscalslidHndl=hndlList(12);
azieditHndl=hndlList(16); eleditHndl=hndlList(18); cscalslidHndl=hndlList(20);
plotHndl=hndlList(24); printHndl=hndlList(33); cbaraxes=hndlList(34);

```

```

% Reset the Handles & variables to their default values:
cla; set(axHndl,'Visible','off','NextPlot','replace')
colormap(jet(64));
rotate3d off
set(startHndl,'Enable','on')
set(demoHndl,'Enable','on')
set(movieHndl,'Enable','on')
set(continHndl,'Value',1,'Enable','off')
set(grppopupHndl,'Value',1,'Enable','off','String','Near-Field Group')
set(fpspopupHndl,'Value',1,'Enable','off')
set(frampopupHndl,'Value',1,'Enable','off','String','Total Frames')
set(edittextHndl,'String','Look Here for All Commands & Prompts!')
set(azieditHndl,'String','-37.5')
set(eleditHndl,'String','30')
text(0,0.1,'Press the "Start" button to begin the analysis,', ...
    'HorizontalAlignment','center');
text(0,0,'Press the "Demo" button to see a demonstration of the software,', ...
    'HorizontalAlignment','center');
text(0,-0.1,'Press the "Movie" button to replay a saved movie or create a new one.', ...
    'HorizontalAlignment','center');
text(0,-0.2,'Or click on the "Plot" menubar to create various plots using data from your last
session.', ...
    'HorizontalAlignment','center');
text(0,-0.4,'All commands will be issued in the text window at the top of this screen.', ...
    'HorizontalAlignment','center');
axis([-1 1 -1 1])
set(plotHndl,'Enable','on')
axes(cbaraxes), cla
set(cbaraxes,'Visible','off'), axes(axHndl)
set(figNumber,'Visible','on', 'UserData',hndlList);

end

```


APPENDIX B. Theoretical Near E-Field Pattern of a Simple Dipole

```
% DipoleEzf.m
%
% Calculates the theoretical z-component of the near electric field of a simple
% dipole transmitting antenna and compares it to the same AWAS-calculated near
% field. The dipole is assumed to be along the z-axis, and measurements are
% taken along the y-axis. Also, the code is optimized for single-frequency
% analysis within a single "near-field group", as defined by AWAS.

clear all

%      *** Reading AWAS Antenna Geometry Data ***

[geofile,filepath] = uigetfile('*.geo','AWAS Geometry Files');
geofid = fopen(geofile,'rt');
geohdr = fscanff(geofid,'%76c',1); % Skips the header data on the .GEO file
g1 = fscanff(geofid,'%3d',1); % Skips the first integer value
GndPlane = fscanff(geofid,'%3d',1); % Indicates presence of a Ground Plane
g2 = fscanff(geofid,'%3d',3); % Skips the next 3 integer values
node1 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #1
node2 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #2
node3 = fscanff(geofid,'%14e',3); % X, Y, & Z coordinates for Node #3

%      *** Reading AWAS Near-Field Data ***

[filename,filepath] = uigetfile('*.nfl','AWAS Near-Field Files');
fid = fopen(filename,'rt');
Ng = fscanff(fid,'%3d',1); % Number of Position Groups
Fp = fscanff(fid,'%12e',2); % (f1, f2) Start/Stop Freq Pts
Ns = fscanff(fid,'%4d',1); % Number of AWAS Freq Steps
f1 = Fp(1)/1e6; f2 = Fp(2)/1e6; df = (f2-f1)/Ns; f = 0:df:f2;
AntennaType = fscanff(fid,'%3d',1); % The Antenna Type (Transmitter/Rcvr)
Nd = fscanff(fid,'%3d',2); % Skips the next 3 integer values.

xp = zeros(Ng,2); yp = zeros(Ng,2); zp = zeros(Ng,2);

for ng=1:Ng; % Reading Group Coordinates
    xp(ng,:) = fscanff(fid,'%12e',[1,2]); % (x1, x2) Near Fld Pts
    Nxs(ng) = fscanff(fid,'%4d',1); % No. x-steps
    yp(ng,:) = fscanff(fid,'%12e',[1,2]); % (y1, y2) Near Fld Pts
    Nys(ng) = fscanff(fid,'%4d',1); % No. y-steps
    zp(ng,:) = fscanff(fid,'%12e',[1,2]); % (z1, z2) Near Fld Pts
```



```

    Nzs(ng) = fscanf(fid,'%4d',1);    % No. z-steps
end

NG = 1;
x1 = xp(NG,1); x2 = xp(NG,2);
y1 = yp(NG,1); y2 = yp(NG,2);
z1 = zp(NG,1); z2 = zp(NG,2);
Nx = Nxs + ones(size(Nxs));  Ny = Nys + ones(size(Nys));
Nz = Nzs + ones(size(Nzs));    % Total # of x,y & z points
if Nxs(NG)==0, dx = 0; else dx = (x2-x1)/Nxs(NG); end, x = x1:dx:x2;
if Nys(NG)==0, dy = 0; else dy = (y2-y1)/Nys(NG); end, y = y1:dy:y2;
if Nzs(NG)==0, dz = 0; else dz = (z2-z1)/Nzs(NG); end, z = z1:dz:z2;

if z1==z2,    % AWAS near-field data is in the xy-plane
    N1 = Nx(NG); N1s = Nxs(NG); N2 = Ny(NG); N2s = Nys(NG);
    rowposit = 1:N2; colposit = (N1+1)/2; xdim = x; xlabl = 'x'; ydim = y;
    ylabl = 'y'; minxdim = x1; maxxdim = x2; minydim = y1; maxydim = y2;
else
    % AWAS near-field data is in the yz-plane
    N1 = Ny(NG); N1s = Nys(NG); N2 = Nz(NG); N2s = Nzs(NG);
    rowposit = (N2+1)/2; colposit = 1:N1; xdim = y; xlabl = 'y'; ydim = z;
    ylabl = 'z'; minxdim = y1; maxxdim = y2; minydim = z1; maxydim = z2;
end
Nf = Ns+2;    % Number of points to f2 folding freq (needs to be 2^m+1)
Nrow = 2*Nx.*Ny.*Nz;  EH = zeros(6,Nrow(NG));
Eyf = zeros(N2,N1,Nf); if GndPlane==1 & z1~=z2, Eyf = zeros(2*N2s+1,N1,Nf); end
Ezf = Eyf;  Hxf = Ezf;

for m=2:Nf;    % Frequency Stepping
    for n=1:NG;    % Reading Group Data
        S = fgetl(fid);    % Read blank line between EH data arrays
        EH = fscanf(fid,'%12e',[6,Nrow(n)]);
        if n == NG,    % Saving Data at Selected Group
            Eyf(1:N2, :,m) = reshape(EH(3,1:2:Nrow(NG)-1)+j*EH(4,1:2:Nrow(NG)-1)',N2,N1);
            Ezf(1:N2, :,m) = reshape(EH(5,1:2:Nrow(NG)-1)+j*EH(6,1:2:Nrow(NG)-1)',N2,N1);
            Hxf(1:N2, :,m) = reshape(EH(1,2:2:Nrow(NG))+j*EH(2,2:2:Nrow(NG))',N2,N1);
        end
    end
end

if GndPlane==1
    Eyf=.5*Eyf; Ezf=.5*Ezf; Hxf=.5*Hxf; % Scale factor to comp for gndplane use
    if z1~=z2,    % Changes the appropriate variables due to

```

```

N2s=2*N2s; N2=N2s+1; z1=-z2; % the presence of the ground plane.
dz=(z2-z1)/N2s; z=z1:dz:z2; rowposit=(N2+1)/2; ydim=z; minydim=z1;
Eyf((N2+1)/2:N2,,:)=Eyf(1:(N2+1)/2,,:); % Reshape the Eyf matrix to
Eyf(1:N2s/2,,:)=flipdim(Eyf((N2+3)/2:N2,,:),1); % fit the gnd plane "image"
Ezf((N2+1)/2:N2,,:)=Ezf(1:(N2+1)/2,,:); % Reshape the Ezf matrix to
Ezf(1:N2s/2,,:)=flipdim(Ezf((N2+3)/2:N2,,:),1); % fit the gnd plane "image"
Hxf((N2+1)/2:N2,,:)=Hxf(1:(N2+1)/2,,:); % Reshape the Hxf matrix to
Hxf(1:N2s/2,,:)=flipdim(Hxf((N2+3)/2:N2,,:),1); % fit the gnd plane "image"
end
end
fclose(fid); fclose(geofid); clear EH

% *** Reading AWAS Input Impedance Data ***

[filename,filepath] = uigetfile('*.zpm','AWAS Z Parameter Files');
Zfid = fopen(filename,'rt');
Np = fscanf(Zfid,'%3d',1); % Number of Ports
Fp = fscanf(Zfid,'%12e',2); % (f1, f2) Start/Stop Freq Pts
Ns = fscanf(Zfid,'%4d',1); % Number of AWAS Freq Steps
zp = zeros(Ns+1,2);
for ns=1:Ns+1; % Reading Group Coordinates
    zp(ns,:) = fscanf(Zfid,'%12e',[1,2]); % (Zreal, Zimag) Zin points
end
Zp = zp(:,1) + j*zp(:,2);
if length(Zp)>1, Zp = [1;Zp]; end

Ez = Zp*Ezf(:,2); % The final AWAS Near-Field Values

% Calculating the theoretical near-field values of Ez
c=3e8; iin=1;
f = input('Enter the frequency (MHz): ');
k=2e6*pi*f/c;
% L = input('Enter the dipole length (meters): ');
L=2.06156;
H=L/2; kH=k*H; coskH=cos(kH); sinkH=sin(kH);
% zpt = input('Enter fixed z for calculation (meters): ');
zpt = 0;
R1=sqrt((zpt-H)^2 + y'.^2); kR1=R1*k;
R2=sqrt((zpt+H)^2 + y'.^2); kR2=R2*k;
R=sqrt(zpt^2 + y'.^2); kR=R*k;

```

```

Ez2=repmat((-j*30*Iin./sinkH),length(y),1).*(exp(-j*kR1)./repmat(R1,1,length(f)) +
exp(-j*kR2)./repmat(R2,1,length(f)) - 2*repmat(coskH,length(y),1).*exp(-
j*kR)./repmat(R,1,length(f)));
figure, plot(y,abs(Ez2(:,1)), 'r--', y, abs(Ez));
xlabel('y (meters)'), title(['Near-Field |Ez| for L= ', num2str(L), ...
' Dipole at z= ', num2str(zpt), ' m & f= ', num2str(f), ' MHz'])
legend('Theoretical Values', 'AWAS Calculations')
figure, plot(y, real(Ez2(:,1)), 'r--', y, real(Ez))
xlabel('y (meters)'), title(['Near-Field Re(Ez) for L= ', num2str(L), ...
' Dipole at z= ', num2str(zpt), ' m & f= ', num2str(f), ' MHz'])
legend('Theoretical Values', 'AWAS Calculations')
figure, plot(y, imag(Ez2(:,1)), 'r--', y, imag(Ez))
xlabel('y (meters)'), title(['Near-Field Im(Ez) for L= ', num2str(L), ...
' Dipole at z= ', num2str(zpt), ' m & f= ', num2str(f), ' MHz'])
legend('Theoretical Values', 'AWAS Calculations')

```

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2. Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3. Director, Training and Education MCCDC, Code C46 1019 Elliot Road Quantico, VA 22134-5027	1
4. Director, Marine Corps Research Center MCCDC, Code C40RC 2040 Broadway Street Quantico, VA 22134-	2
5. Director, Studies and Analysis Division MCCDC, Code C45 3300 Russell Road Quantico, VA 22134-5130	1
6. Marine Corps Tactical Systems Support Activity Technical Advisory Branch Attn: Maj J.C. Cummiskey Box 555171 Camp Pendleton, CA 92055-5080	1
7. Marine Corps Representative Naval Postgraduate School Code 037, Bldg. 234, HA-220 699 Dyer Road Monterey, CA 93940	1

8. Chairman, Code EC 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121
9. Professor Michael A. Morgan, Code EC/Mw 2
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121
10. Professor R. Clark Robertson, Code EC/Rc 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121
11. Maj David V. Adamiak, Code EC/Ad 2
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121